TP 3097

/MM

Transport Canada / Transports Canada
Library and Information Centre /
Centre de Documentation G S L A
Ottawa, Canada K1A 0N5

# M.V. ARCTIC
# INSTRUMENTATION AND DATA COLLECTION SYSTEM

# FOR

# SHIP OPERATIONS IN ICE

**

## SYSTEM OPERATING AND USER'S MANUAL

## VOLUME III
## (SOFTWARE MANUAL)

FENCO CONSULTANTS LTD.
in conjunction with
GERMAN & MILNE INC.

MAY 1981

# REPORT DOCUMENTATION FORM

1. Transport Canada Report No.
TP 3097

2. TDC Project No.
4127

3. Recipient's Catalogue No.

4. Title and Subtitle
M.V. "Arctic" Instrumentation and Data Collection
System for Ship Operations in Ice
System Operating and User's Manual
Volumes 1, 2 and 3

5. Report Date
May 1981

6. Performing Organization Report No.

7. Author(s)
D.J. Boteler, A. Benjamin & J.G. German

8. Transport Canada File No.
D 500-102-13

9. Performing Organization Name and Address
Fenco Consultants Limited,
9th Floor, Lavalin Centre
909, Fifth Avenue S.W.
Calgary, Alberta T2P 3G5

10. DSS File No.
09SD. T8200-9-9546

11. DSS or Transport Canada Contract No.
05D80-00010

12. Sponsoring Agency Name and Address
Transportation Development Centre (TDC)
1000, Sherbrooke Street West
P.O. Box 549
Montreal, Quebec H3A 2R3

13. Type of Report and Period Covered
Final - Dec.1980-May 1981

14. Sponsoring Agency Code

15. Supplementary Notes
This report supercedes TP 2217, Vol. 1 & 2

16. TDC Project Officer
D. Hearnshaw

17. Abstract

The report provides an updated description of an Instrumentation and Data Collection System installed in the motor vessel "Arctic". This system is designed to provide a record of the ship's operational experience when navigating in ice, both in terms of the prevailing environmental conditions and the ship's performance and response. Records of hull strains resulting from ice impacts, together with ship machinery parameters (such as shaft RPM, fuel rack settings, torque, thrust and propeller pitch) and ship speed, direction and motions are obtained with this system. A computer-controlled time-division-multiplex system is used to sample the analogue output of each sensor, convert to digital form, and record on magnetic tape. Recording can be started manually or set to start automatically when specified signals exceed preset threshold levels. This report contains complete user information for operating the system and for carrying out routing hardware maintenance and software changes.

Volume 1 - Provides a general description of the system with operating and calibrating instruction, and a discussion of sampling schemes.
Volume 2 - Provides a detailed description of the hardware components including the CCTV System, along with manufacturer's information.
Volume 3 - Provides an introduction to the Rolm 1603A Computer, with details of the data acquisition and playback programs.

18. Key Words
Instrumentation
Ice performance
Icebreaking
Arctic vessels
Hull

19. Distribution Statement
Limited number of copies available from TDC.

20. Security Classification (of this report)
NIL

21. Security Classification (of this page)
NIL

22. No. of Pages
LOOSE-LEAF

23. Price

TDC CDT 79-005

# FORMULE DE DOCUMENTATION POUR RAPPORT

1. N° du rapport Transports Canada
TP 3097

2. N° de l'étude — CDT
4127

3. N° de catalogue du destinataire

4. Titre et sous-titre
M.V. "Arctic" Instrumentation and Data Collection
System for Ship Operations in Ice
System Operating and User's Manual
Volumes 1, 2 and 3

5. Date du rapport
mai 1981

6. N° du rapport de l'organisme exécutant

7. Auteur(s)
D.J. Boteler, A. Benjamin & J.G. German

8. N° de dossier Transports Canada
D 500-102-13

9. Nom et adresse de l'organisme exécutant
Fenco Consultants Limited
Lavallin Centre, 9ième étage
909, Fifth Avenue, S.W.
Calgary, Alberta T2P 3G5

10. N° de dossier — ASC
09SD.T8200-9-9546

11. N° de contrat — ASC ou Transports Canada
05D80-00010

12. Nom et adresse de l'organisme parrain
Centre de développement des transports
1000 ouest, rue Sherbrooke
C.P. 549
Montréal, Québec H3A 2R3

13. Genre de rapport et période visée
Final - déc. 1980-mai 1981

14. Code de l'organisme parrain

15. Remarques additionnelles
Ce rapport mis à jour remplace le rapport TP 2217, volumes 1 et 2.

16. Agent de projet — CDT
D. Hearnshaw

17. Résumé

Ce rapport constitue une description mise à jour de l'instrumentation et du système de saisie de données installés à bord du cargo brise-glace Arctic et destinés à relever le comportement du navire dans les glaces en enregistrant les conditions rencontrées de navigation et les caractéristiques correspondantes de manoeuvre. Les déformations de la coque sous l'impact des glaces et les paramètres du fonctionnement de la machinerie (révolutions de l'arbre en tr/min., réglage des injecteurs de carburant, couple moteur, poussée et pas d'hélice), ainsi que la vitesse, la direction et les mouvements du navire ont été relevés. Les signaux des jauges de contrainte, sortant en analogie, sont enregistrés par un ordinateur qui en compile l'échantillonnage et le multiplexage par répartition dans le temps, les convertit en numérique et les consigne sur bande magnétique. L'enregistrement peut être déclenché soit manuellement soit automatiquement et, dans ce dernier cas, dès que certains signaux dépassent un seuil fixé à l'avance. Ce rapport fournit aussi des instructions complètes aux usagers sur la mise en oeuvre du système en question, sur l'entretien normal du matériel et sur la manipulation des logiciels.

Volume 1 - Description générale du système de saisie de données, instructions sur la mise en oeuvre et l'étalonnage, et suggestions sur les modalités d'échantillonnage.

Volume 2 - Description détaillée des composants matériels, y compris la télévision en circuit fermé, et instructions du manufacturier.

Volume 3 - Description de l'ordinateur Rolm 1603A et instructions détaillées sur l'acquisition et la lecture des données.

18. Mots-clés
Instrumentation
Comportement dans les glaces
Déglaçage
Navires de catégorie arctique
Coque

19. Diffusion
Le CDT dispose d'un nombre limité d'exemplaires.

20. Classification de sécurité (de ce rapport)
NIL

21. Classification de sécurité (de cette page)
NIL

22. Nombre de pages
Feuilles mobiles

23. Prix

CDT TDC 79-005

# SYSTEM OPERATION AND USER'S MANUAL

## VOLUME I

## CONTENTS

PART I: GENERAL DESCRIPTION OF INSTRUMENTATION AND DATA COLLECTION SYSTEM

Section 1: Introduction
Section 2: Objectives
Section 3: Data Collection System

PART II: OPERATING INSTRUCTIONS

Section 1: Introduction
Section 2: System Operation
Section 3: Data Acquisition Initialization
Section 4: Playback
Section 5: Test Modes
Appendix A: An Introduction to Octal Representation
Appendix B: Alphabetical Listing and Binary Addresses of Sensor

PART III: CALIBRATION AND SETTING-UP PROCEDURES

Section 1: Introduction
Section 2: Amplifier Cards
Section 3: Master Cards
Section 4: Interface Buffer

PARTS IV: DISCUSSION OF SAMPLING SCHEMES

Section 1: Introduction
Section 2: Aliasing and the Need for Filtering
Section 3: Filters Available
Section 4: Strain Gauges
Section 5: Ship Performance Sensors
Section 6: Environmental Sensors
Section 7: Propulsion Sensors

# SYSTEM OPERATION AND USER'S MANUAL

## VOLUME II

## CONTENTS

PART I: GENERAL DESCRIPTION OF EQUIPMENT

PART II: DETAILED DESCRIPTION OF EQUIPMENT

- Section 1: Hull Structure Sensors
- Section 2: Ship Performance Sensors
- Section 3: Propulsion Sensors
- Section 4: Bubbler Sensors
- Section 5: Environment Sensors
- Section 6: Subjunction Boxes and Data Bus
- Section 7: Interface
- Section 8: Computer System

PART III: CLOSED CIRCUIT T.V. SYSTEM

PARTS IV: MANUFACTURER'S INFORMATION

# SYSTEM OPERATION AND USER'S MANUAL

## VOLUME III

## CONTENTS

PART I: INTRODUCTION

PART II: DATA ACQUISITION PROGRAM

PART III: PLAYBACK PROGRAM

APPENDIX A: Data Acquisition Program Listing

B: Playback Program Listing

C: Software Back-Up

D: System Diagnostics

E: ROLM Documentation

# PREFACE

This manual describes software developed for the "M.V. Arctic Data Acquisition System". It is intended as an aid to the System Programmer performing software maintenance operations on existing software. The manual should be used in conjunction with 1603A ROLM Corporation Software Documentation provided.

A general introduction to the ROLM 1603A Computer is presented in Section I, with special reference to the hardware configuration in use. Programming for the Data Acquisition program is discussed in Section II, while that for the Playback Program in Section III. Listings for both the programs may be found in the Appendices.

It has been presumed in all discussions on system software that the reader is already familiar with the operation of the Data Acquisition System (see Volume I of M.V. Arctic, Instrumentation and Data Collection System for ship operations in Ice - System Operating and User's Manual).

TABLE OF CONTENTS

Page No.

1. INTRODUCTION 1
   - 1.1 Computer Overview 1
   - 1.2 System Description 1
   - 1.3 System Software 3
   - 1.3.1 Operating Systems 3
   - 1.3.2 Programming Languages 3
   - 1.3.3 Utility Routines 4
   - 1.3.4 Interrupts 4
   - 1.3.5 Power Failure 4
   - 1.3.6 Peripherals 4
   - 1.3.7 Debugger 5
   - 1.4 Data Gathering System 5

2. DATA ACQUISITION PROGRAM 7
   - 2.1 Introduction 7
   - 2.2 Program Modules 8
   - 2.3 Task Allocation 14
   - 2.4 Stack Structure 20
   - 2.5 Utility Subroutines 24
   - 2.6 Magnetic Tape Formats 30
   - 2.7 Program Structure Outline 36
   - 2.8 Conversion Equations 44
   - 2.9 Creating a Save File (MV.SV) 47

3. PLAYBACK PROGRAM 54
   - 3.1 Introduction 54
   - 3.2 Program Modules 54
   - 3.3 Task Allocation 60
   - 3.4 Program Structure Outline 63
   - 3.5 Creating a Save File (PB.SV) 68

## APPENDICES

Page No.

Appendix A - Data Acquisition Program Listing

A.1 MV0. A-1
A.2 MV1. A-13
A.3 MV2. A-25
A.4 MV3. A-35
A.5 MV8. A-47
A.6 MVPV. A-57
A.7 MVDT. A-94
A.8 MVSP. A-92
A.9 MVAL. A-109
A.10 MVLG. A-117
A.11 MVMN. A-126
A.12 MVRM. A-133
A.13 MVTR. A-138
A.14 MVMT. A-151
A.15 MVVB. A-165

Appendix B - Playback Program Listing

B.1 PB .FR B-1
B.2 PB2DT.FR B-4
B.3 PBACC.FR B-5
B.4 PBDP .FR B-6
B.5 PBESC.FR B-7
B.6 PBKES.FR B-8
B.7 PBMTH.FR B-9
B.8 PBMTR.FR B-10
B.9 PBMTS.FR B-11
B.10 PBALM.FR B-12
B.11 PBMN. FR B-14
B.12 PBMNF.FR B-17
B.13 PBSTG.FR B-18
B.14 PBSTI.FR B-19
B.15 PBVAL.FR B-20
B.16 PBSTT.FR B-24
B.17 IPBYNR.FR B-27

Appendix C - Software Back-Up C-1

Appendix D - System Diagnostics D-1

Appendix E - ROLM Documentation E-1

## LIST OF FIGURES

Page No.

Figure 1 ROLM 1603 Block Diagram 2
Figure 2 Software Overview 6
Figure 3 Data Acquisition Program - Overview 9
Figure 4 Data Acquisition Program - Interrupt Driven Processing 10
Figure 5 Data Acquisition Program - Task Allocation 15
Figure 6 Playback Program - Overview 55
Figure 7 Playback Program - Task Interaction 61

# CHAPTER 1

# INTRODUCTION

# 1. INTRODUCTION

## 1.1 Computer Overview

The ROLM[1] Model 1603A computer is 16-bit general purpose computer designed and built for reliable operation in severe environments. The computer has 32K words of core memory, which can be extended to 64K words. Options are available for a variety of input/output (I/O) equipment including parallel and serial digital interfaces, analog-to-digital converters, and serial communication interfaces. Controllers are available for a wide range of peripheral devices, including terminal printers, paper tape equipment, line printers, magnetic tape, and disk.

## 1.2 System Description

A simplified block diagram of the processor is given in Figure 1. It shows the basic interrelation between the processor assemblies. The CPU provides for all transfer and manipulation of data. Words (data and instructions) can be stored or fetched from memory. The processor executes a program by retrieving instructions from memory locations, under control of the Program Counter, PC. At the end of each instruction PC is incremented by one, so that the next instruction is normally taken from the next consecutive location. Sequential program flow is altered by changing the contents of PC, either by incrementing it an extra time in a test-skip instruction or by replacing its contents with the value specified by a jump instruction. The other internal registers of importance to the programmer are four 16-bit accumulators, AC0 through AC3. Data can be moved in either direction between any memory location and any accumulator. Associated with the accumulators is the Carry flag. Carry indicates when a carry occurs out of bit 0 in an arithmetic instruction. Operands for arithmetic and logical operations may be located in the accumulators or, in some cases, in memory. Arithmetic and logical instructions include AND, add, subtract, complement, negate, and right or left shift. Memory reference instructions allow use of AC2 or AC3 as an index register, as well as multiple-level indirect addressing.

The input/output hardware allows the program to address up to 61 devices. A single instruction can transfer a word between an accumulator and a device interface and at the same time control the device operation. Included in the I/O system are facilities for program interrupts and high-speed data transfers. The interrupt system facilitates processor control of the peripheral equipment by allowing any device to interrupt the normal program flow on a priority basis. A high-speed device, such as a magnetic tape unit or disk drive, can gain direct access to memory through a data channel without execution of any instructions. The data channel logic allows transfer of data to or from memory, or incrementing of a memory word. All communications with external devices are performed

---

[1] ROLM is a registered trademark of ROLM Corporation, Santa Clara, California, U.S.A.

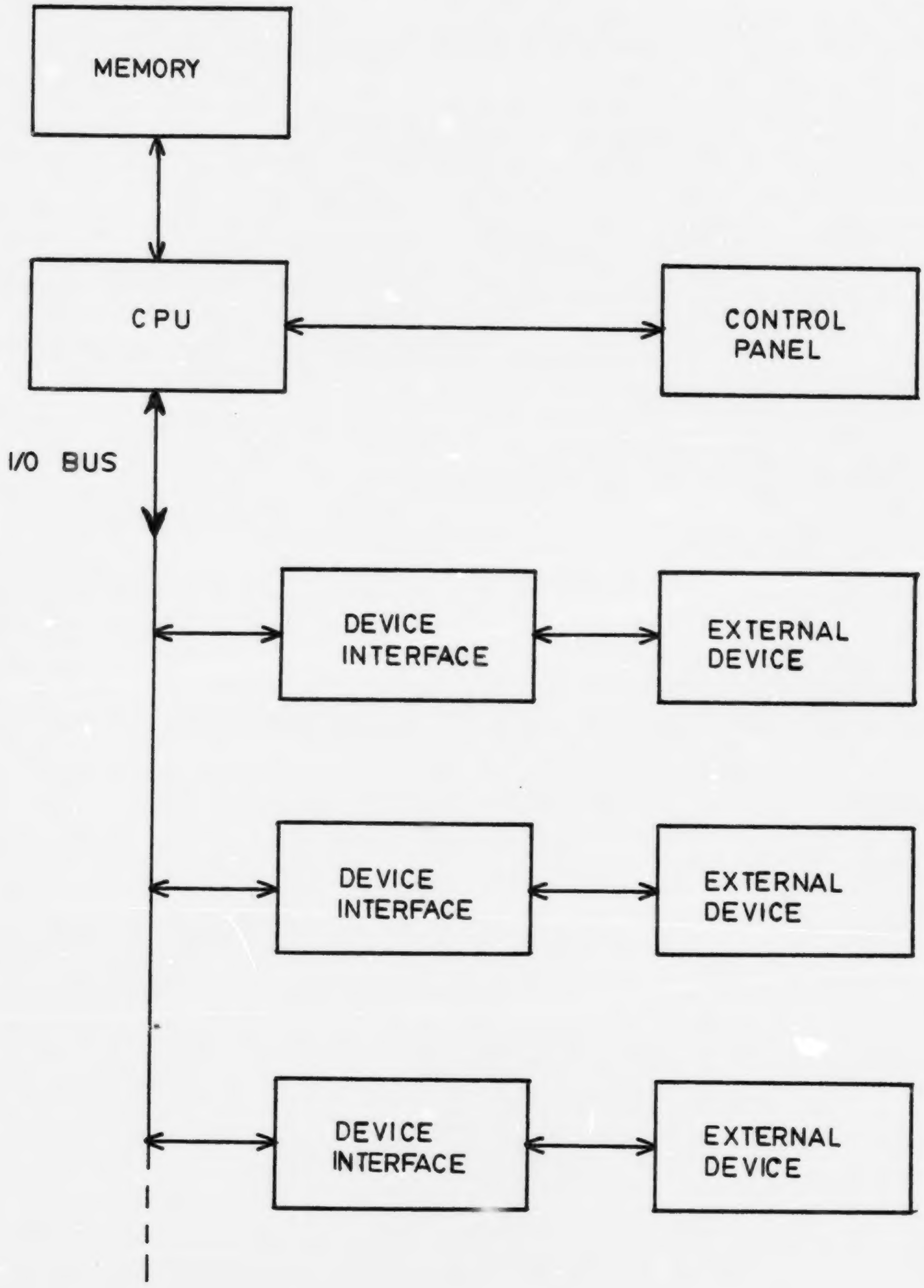
MEMORY
CPU
CONTROL PANEL
I/O BUS
DEVICE INTERFACE
EXTERNAL DEVICE
DEVICE INTERFACE
EXTERNAL DEVICE
DEVICE INTERFACE
EXTERNAL DEVICE

FIGURE 1 -ROLM 1603 BLOCK DIAGRAM

through the I/O-Bus. This bus consists of 16 bidirectional data lines, 6 device selection lines, and 26 control lines. Since external devices operate with different control requirements and data transfer rates, a device interface is generally required to adapt the device to the I/O-Bus circuits and timing.

The control panel provides data entry and control switches. It includes indicators to display the contents of a memory location or accumulator. The program counter and various flags are also displayed.

### 1.3 System Software

The 1603 is supported by a wide range of software which, in turn, supports all of ROLM's peripheral devices. Several different operating systems are available for use with the 1603 depending on the user's requirements. Assembly and high level languages (FORTRAN IV, ALGOL, and BASIC) are available for use with the selected operating system. Normal software development tools (editors, debuggers, and assemblers) are also provided.

#### 1.3.1 Operating Systems

** RDOS (Real-Time Disk Operating System) supports all major peripherals and support programs, including high level language compilers. It is used to perform Software maintenance both on programs for data acquisition as well as those for data analysis.

** RTOS (Real-Time Operating System) consists of a small, memory-resident, general-purpose multitask monitor designed to control a variety of real-time environments. It is a fully compatible subset of RDOS. Programs for both data acquisition and data analysis are designed to run under RTOS.

#### 1.3.2 Programming Languages

** Real-time FORTRAN IV includes all standard ANSI FORTRAN features, plus a number of extensions. FORTRAN IV is suitable for real-time usage. Programs can run in a multitasking environment under RTOS and under RDOS. Under RDOS, full program chaining and overlay operations are allowed. FORTRAN IV has re-entrant libraries, sophisticated file structuring with flexible run-time assignments to I/O devices and a variety of accessing privileges.

Among the high level languages provided, only FORTRAN IV is used in the M.V. Arctic software. ALGOL and BASIC however, are also available for use on this System.

** ALGOL. The ALGOL compiler is a superset of ALGOL 60 with extensions permitting free form I/O or formatted output, recursive and re-entrant procedures, and many other extensions.

** BASIC. BASIC is a high level, core-resident interpreter. It provides rapid and easy programming capability.

While economics obtained from the use of high level languages in applications programs typically reduce development costs by approximately 45%, they increase core-requirements by 20% and program execution time by up to 45%.

Thus, to obtain a higher throughput, the data acquisition software was developed in assembler language so as to operate at the maximum execution speed of the computer. For analysis of the data, the sophistication of FORTRAN IV with its inherent programming power was preferred.

### 1.3.3 Utility Routines

Utility routines available are:

- extended assembler
- macro assembler
- relocatable loader
- text editor
- library file editor
- symbolic debugger
- diagnostic software

### 1.3.4 Interrupts

Many input/output devices must be serviced infrequently relative to the processor speed, but within a short time after they request it. Failure to service a device within the specified time (which varies among devices), can result in loss of information. The program interrupt is designed with these considerations in mind. Interrupt sequences designed for the data acquisition programs facilitate concurrent operation of the main program and a number of peripheral devices.

### 1.3.5 Power Fail

Within the CPU there is logic capable of early detection of AC power failure. The power fail interrupt has the highest priority (00) and cannot be masked or cleared. There is a 500 microsecond delay before the processor stops. This is a significant feature of the system operating under the fluctuating voltage conditions of the ship's power distribution system.

### 1.3.6 Peripherals

Peripheral devices used in this System are:

| | |
|---|---|
| a) Ampex | Magnetic Tape |
| b) AED | Floppy Disk |
| c) G & M | Main Interface to System Sensors |

The device interfaces are connected to the I/O Bus, which is used for all communication between the device and the processor. Each device responds only to programmed I/O instructions containing that particular device's address. The device makes sure of two status flags: Busy and Done. These flags control the activity of the device interface.

### 1.3.7 Debugger

The Symbolic Debugger (RTDEB03) interfaces with user routines, allowing the programmer to monitor and correct the program during execution. It allows the programmer to set up breakpoint locations within the program at which execution of the program will halt. By issuing debugger commands the programmer can examine the contents of memory locations and special registers such as the Accumulators and Carry and can correct the contents if necessary. As the debugger is symbolic, the contents may be examined in source language format, octal format, or a number of special formats. The programming can restart execution at any other location in the program.

## 1.4 Data Gathering System

Two programs have been developed for the M.V. "Arctic" Instrumentation System:

a) Data Acquisition Program
b) Playback Program

A general overview of the System Software is shown in Figure 2. The required program is loaded into memory from a program tape. The Acquisition program stores information obtained from sensor interrogation onto a data tape in addition to providing either a monitoring or a logging printout. This is determined by an operator communication through the keyboard input. The Playback Program reads the stored information off the data tape providing a statistics printout and also an optional dump or monitoring printout. The programs are discussed in detail in Chapters 2 and 3. Descriptions of use of the programs may be found in Volume I of the System Operating and User's Manual for the Instrumentation and Data Collecting System for Ship's operations in Ice.

# CHAPTER 2

# DATA ACQUISITION PROGRAM

## 2. DATA ACQUISITION PROGRAM

### 2.1 Introduction

The Data Acquisition Program is a real-time assembler program that performs the function of interrogating selected sensors in the system, each at a predetermined rate. It takes advantage of multi-tasking and interrupt techniques to form an effective data gathering tool. It is fully automatic during execution, yet allowing the operator to intervene and gain control at any time. Features of the Data Acquisition Program include:

i) Date and Time

The program keeps track and records the date and time of all recording operations. In addition, the start and stop times of file recording are reported to the operator on the terminal printer.

ii) Sampling Pattern Selection

The operator has complete flexibility over the choice of the sampling pattern. The only constraint is the overall sensor sampling rate which must remain within the system maximum handling capacity of 1765 samples/second. The program will disallow any attempt to exceed this maximum rate. Different sampling rates, may be selected for each sensor. The actual sampling sequence is determined by rate tables that manipulate stack pointers to ensure each sensor is sampled at the proper rate. The available sampling frequences are:

1) 0.1 Hz
2) 1 Hz
3) 10 Hz
4) 50 Hz
5) 100 Hz
6) 500 Hz
7) 1000 Hz

Further, one of two sampling modes may be selected for each sensor:

R : Real (records value at sampling instant)
P : Peak (records peak or maximum value attained since last sample)

When a sampling pattern is to be frequently used, it can be made part of the program and accessed when required. In this case initialization of the sampling pattern is selected after program load. Alternately the programmer may modify the stored sampling pattern or when necessary, the initialized pattern.

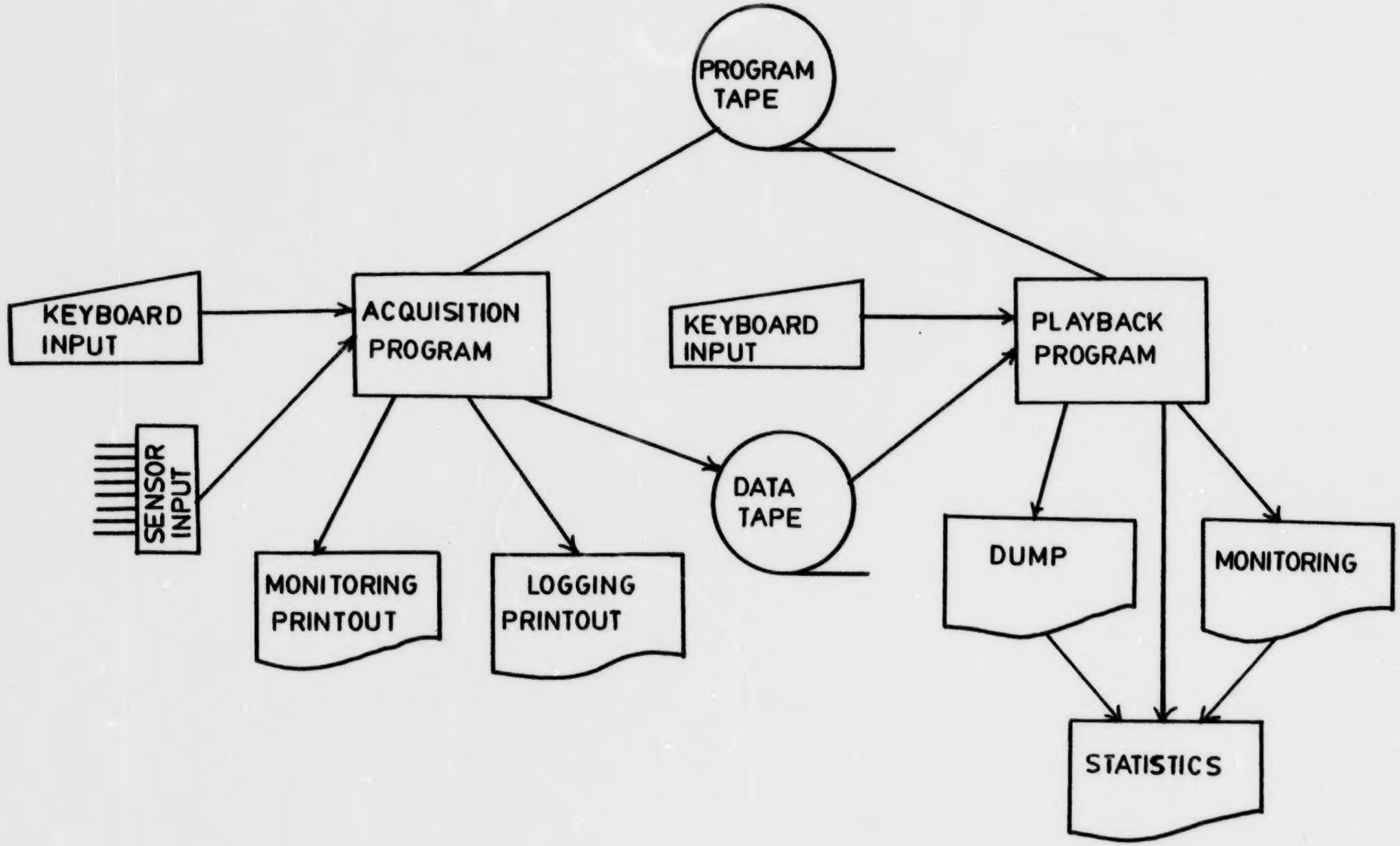
PROGRAM TAPE
KEYBOARD INPUT
ACQUISITION PROGRAM
KEYBOARD INPUT
PLAYBACK PROGRAM
SENSOR INPUT
DATA TAPE
MONITORING PRINTOUT
LOGGING PRINTOUT
DUMP
MONITORING
STATISTICS

FIGURE 2 - SOFTWARE OVERVIEW

iii) Alarms

Alarms are set on selected sensors when the system operates in the Standby Mode. The maximum and minimum alarm values are entered as numbers between 0 and 1023. The corrosion from engineering units must be done manually.

iv) Logging interval

While recording, the program will periodically report the status of each sensor being sampled, via the terminal. The logging interval is decided by the operator. No logging printout is obtained while monitoring.

v) Monitoring

In the monitoring mode only one sensor is continually monitored, and the sampled value printed out on the terminal device. This mode is useful for verification or re-calibration of sensors.

Note: No printout is obtained when the input voltage is outside the ±10 Volt range.

vi) Alarm B

Using this function the program may be set up so as to monitor only one channel when the sampled value exceeds the pre-set alarm.

vii) Recording Mode

There are two modes. In the Recording Mode data are written on tape continuously according to the sampling pattern. In the Standby Mode, recording takes place only when the alarm is exceeded. The duration of this recording interval is predetermined by the operator.

viii) Tape Units

For long periods of unattended recording, the program can be instructed to record two tapes consecutively. On completion of one tape the program automatically transfers to the other tape drive with minimal loss of data.

### 2.2 Program Modules

An overview of program execution is shown in Figures 3 and 4. Multitasking techniques are used which will be discussed in detail in Section 2.3. A user clock provides timing for the interrupts, and a stack is used to sequence sensors to be sampled. Programming has been carried out in modules as listed below.

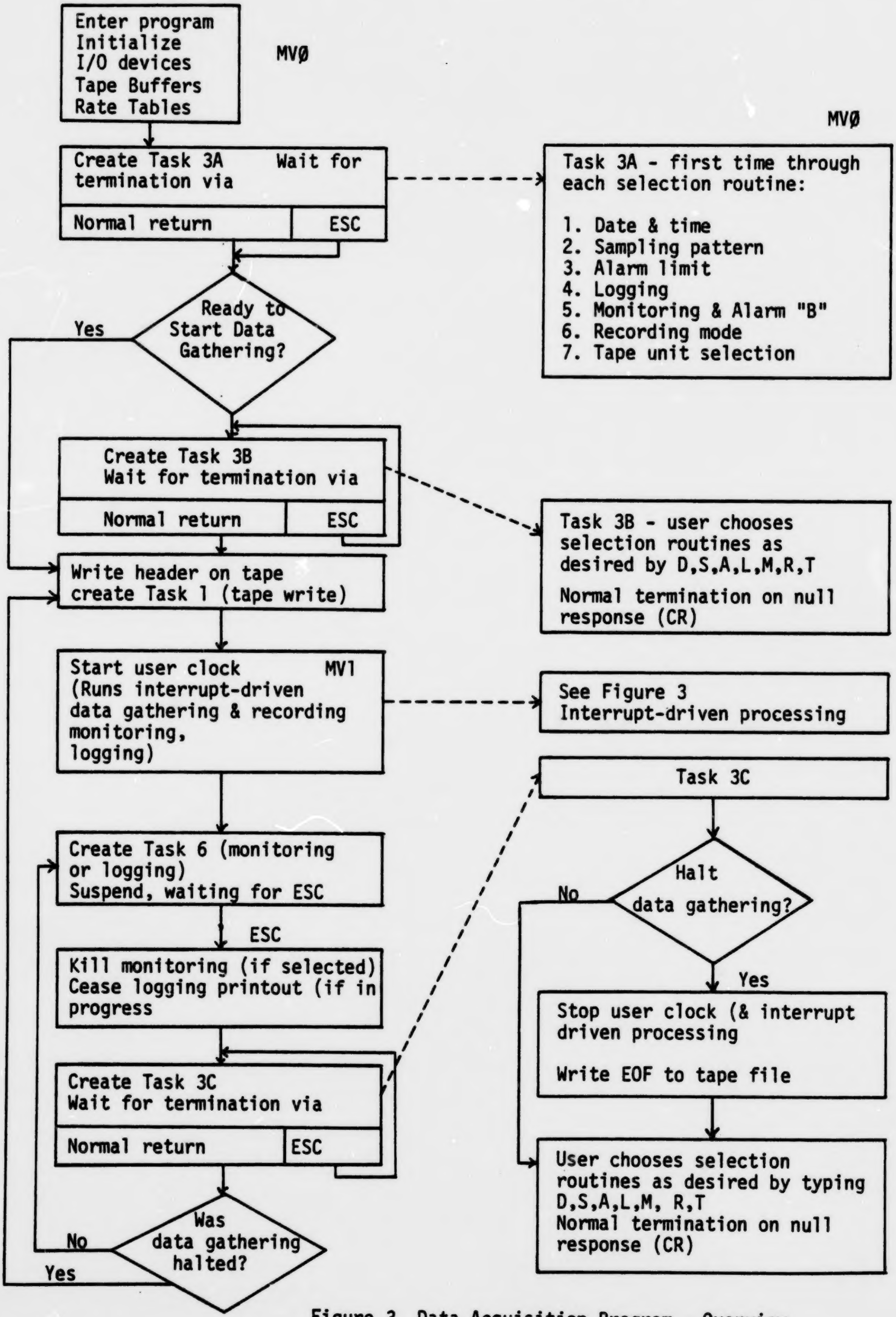
Enter program
Initialize
I/O devices
Tape Buffers
Rate Tables
MV0
MV0
Create Task 3A Wait for termination via
Normal return
ESC
Task 3A - first time through each selection routine:
1. Date & time
2. Sampling pattern
3. Alarm limit
4. Logging
5. Monitoring & Alarm "B"
6. Recording mode
7. Tape unit selection
Yes
Ready to Start Data Gathering?
Create Task 3B
Wait for termination via
Normal return
ESC
Task 3B - user chooses selection routines as desired by D,S,A,L,M,R,T
Normal termination on null response (CR)
Write header on tape
create Task 1 (tape write)
Start user clock
MV1
(Runs interrupt-driven data gathering & recording monitoring, logging)
See Figure 3
Interrupt-driven processing
Task 3C
No
Halt data gathering?
Yes
Create Task 6 (monitoring or logging)
Suspend, waiting for ESC
ESC
Kill monitoring (if selected)
Cease logging printout (if in progress
Stop user clock (& interrupt driven processing
Write EOF to tape file
Create Task 3C
Wait for termination via
Normal return
ESC
User chooses selection routines as desired by typing D,S,A,L,M, R,T
Normal termination on null response (CR)
No
Was data gathering halted?
Yes

Figure 3 Data Acquisition Program - Overview

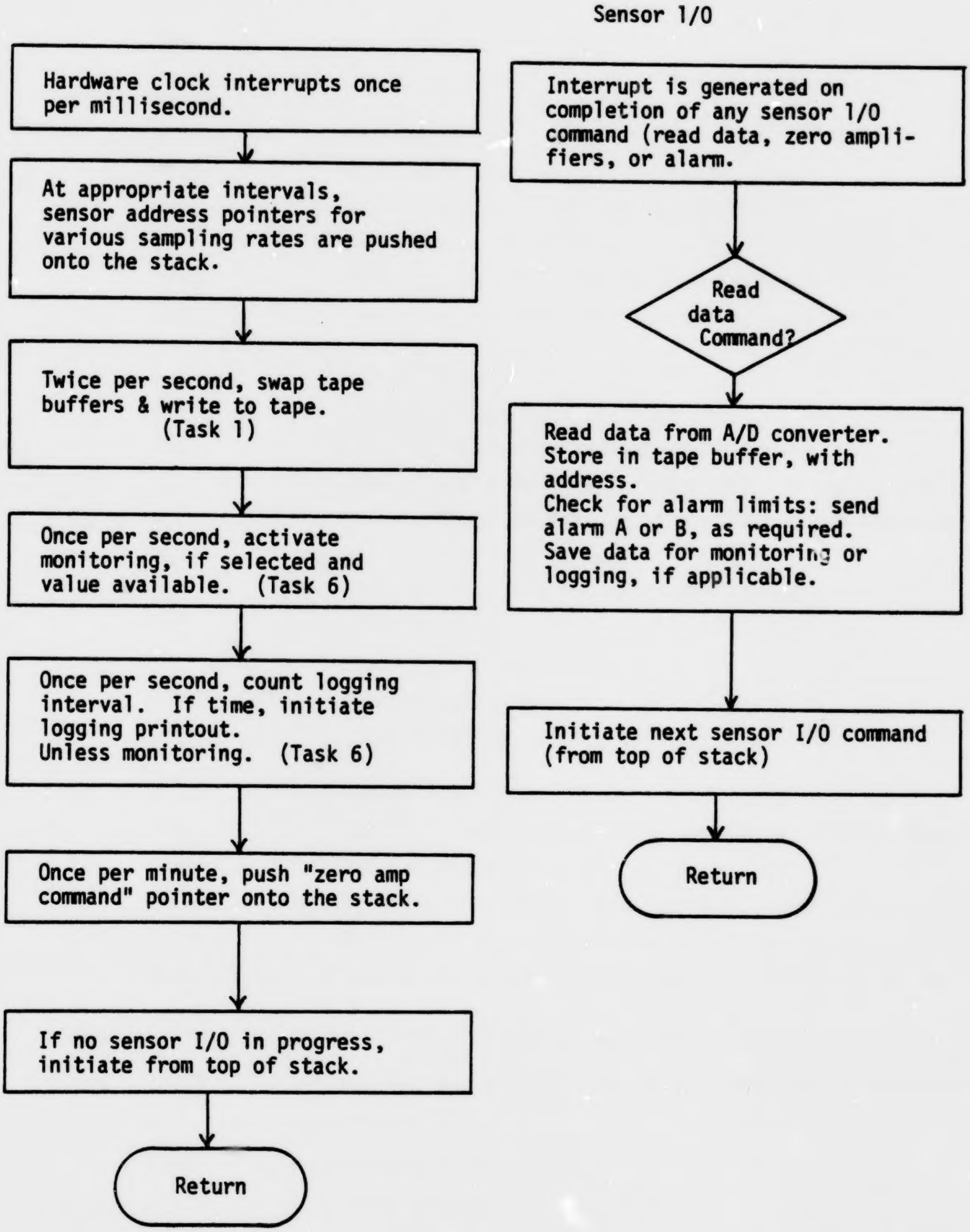
Sensor 1/0
Hardware clock interrupts once per millisecond.
At appropriate intervals, sensor address pointers for various sampling rates are pushed onto the stack.
Twice per second, swap tape buffers & write to tape. (Task 1)
Once per second, activate monitoring, if selected and value available. (Task 6)
Once per second, count logging interval. If time, initiate logging printout. Unless monitoring. (Task 6)
Once per minute, push "zero amp command" pointer onto the stack.
If no sensor I/0 in progress, initiate from top of stack.
Return
Interrupt is generated on completion of any sensor 1/0 command (read data, zero amplifiers, or alarm.
Read data Command?
Read data from A/D converter. Store in tape buffer, with address. Check for alarm limits: send alarm A or B, as required. Save data for monitoring or logging, if applicable.
Initiate next sensor I/O command (from top of stack)
Return

Figure 4 Data Acquisition Program
Interrupt - Driven Processing

## Modules

| | | |
|---|---|---|
| MVO | Task 0 | Initialization/main control |
| | Task 3 | User intervention control |
| MV1 | Constants, tables | |
| | UCLST | User clock start routine |
| | SPUSH | Stack push routine |
| | UCLIS | User clock interrupt service routine |
| | INST | Interface start routine |
| | INIS | Interface in interrupt service routine |
| | OUTIS | Interface out interrupt service routine |
| MV2 | OCASC | Convert binary to octal ASCII |
| | BNASC | Convert binary to decimal ASCII |
| | GTCH | Get next character from an ASCII string |
| | DIG | Convert ASCII decimal to binary (arbitrary length) |
| | DIGN | Convert ASCII decimal to binary (exactly N characters) |
| | RLN | Read a line from terminal |
| | PCR | Print a line with pad character at start |
| | PNL | Print a line |
| | MULT | Integer multiply |
| | DIVID | Integer divide |
| | CVLT | Convert integer binary to ASCII real volts |
| MV3 | GADR | Decode ADR or ADR-ADR or ADR-ADR/INCR |
| | RPS | Report present status of an adr.(rate, mode, alarm,limit) |
| | CKCR | Get and check CR at end of input line |
| | FMTER | Print "FORMAT ERROR" and increment error count |
| WRNAB | WRNAB | Print warning message/offer change to abort |
| | PERR | Type error messages, zero counters at end of a tape file |
| PVAL | Convert and print binary from ADC to engineering units | |
| PVAL | (MVPV) | Convert and print binary output from ADC to engineering units. |
| | CANS | Convert answer to ASCII |
| | STBT | Store a byte |

Modules (Cont'd)

| | | |
|---|---|---|
| | CONVERT | Subroutines to convert 0 thru 1023 to linear functions |
| | TSIN | Table of Line Values |
| | TEXT | Messages to operator |
| MV8 | Test Modules | |
| | TEST | Scope loops |
| | MV800 | Option to print rather than write to tape |
| | TT110 | Test mag tape functions |
| MVVB | Helpful Printout (verbose) Module | |
| MVDT | Date and Time Module | |
| | DT000 | Get date and time from keyboard |
| | DT100 | Print date and time |
| | PDAY | Print date |
| | PTIM | Print time |
| | PDATM | Print date and time |
| | GTDT | Get and convert date |
| | GTTM | Get and convert time |
| MVSP | Sampling Pattern Module | |
| | SP000 | Initialize, modify or check initialization |
| | SP100 | Loop to modify sampling pattern |
| | SP400 | Compute number of samples/second to go |
| | SP450 | Print number of samples/second to go |
| | SP800 | Sort addresses by rate option |
| MVAL | Alarm limits module | |
| | AL000 | Initialization: Initialize, modify or check |
| | AL100 | Loop to modify alarm limits |
| MVLG | Logging Module | |
| | LG000 | Set logging interval from keyboard |
| | LG100 | Print present interval |
| | LG500 | Print logging output TASK 6 |

Modules (Cont'd)

MVMN Monitoring Module

| | |
|---|---|
| MN000 | Set monitoring & alarm "B" addresses from keyboard |
| MN400 | Common processing for monitoring AND ALARM "B" |
| MN100 | Print present address |
| MN500 | Print monitoring output TASK 6 |

MVRM Recording Mode Module

| | |
|---|---|
| RM000 | Set recording mode from keyboard |
| RM100 | Report present mode (recording or standby) |

MVTR Tape Routines Module

| | |
|---|---|
| TR000 | Set up tape drive(s) with keyboard input (initial setup) |
| TR100 | Operator requested tape setup |
| TR500 | Print status of tape drives |
| TR200 | Write out tape buffer to tape: TASK 1 |
| TR240 | EOT processing (new tape) |
| TR300 | Task management for tape drive swap |
| TR400 | EOT processing (old tape): TASK 4 |
| MTEND | Alarm "C" |

MVMT Mag Tape Utilities

| | |
|---|---|
| MTASG | Assign a tape drive; init & open it |
| MTDV | Get and check number of mag tape devices (1 or 2) |
| MTUN | Get and check mag tape unit number (0 or 1) |
| MTHED | Build and write tape header |
| MTDER | Process mag tape errors |
| MTEF | Write 2 end of file marks |
| MTINI | Initialize and open mag tape |
| MTXER | Type tape error messages |
| MTREL | Close and release mag tape |

## 2.3 Task Allocation

Full advantage has been taken of the software multitasking capability to allow efficient sharing of the system resources. Each task may be independent of and occur asynchronously with other tasks. Tasks compete simultaneously for the use of system resources in a multitasking environment, although only one task may reserve CPU control and the desired resources at a given instant. The task scheduler allocates control to each task, based on task priority and readiness to use system resources. A diagram of the task set-up is shown in Figure 5.

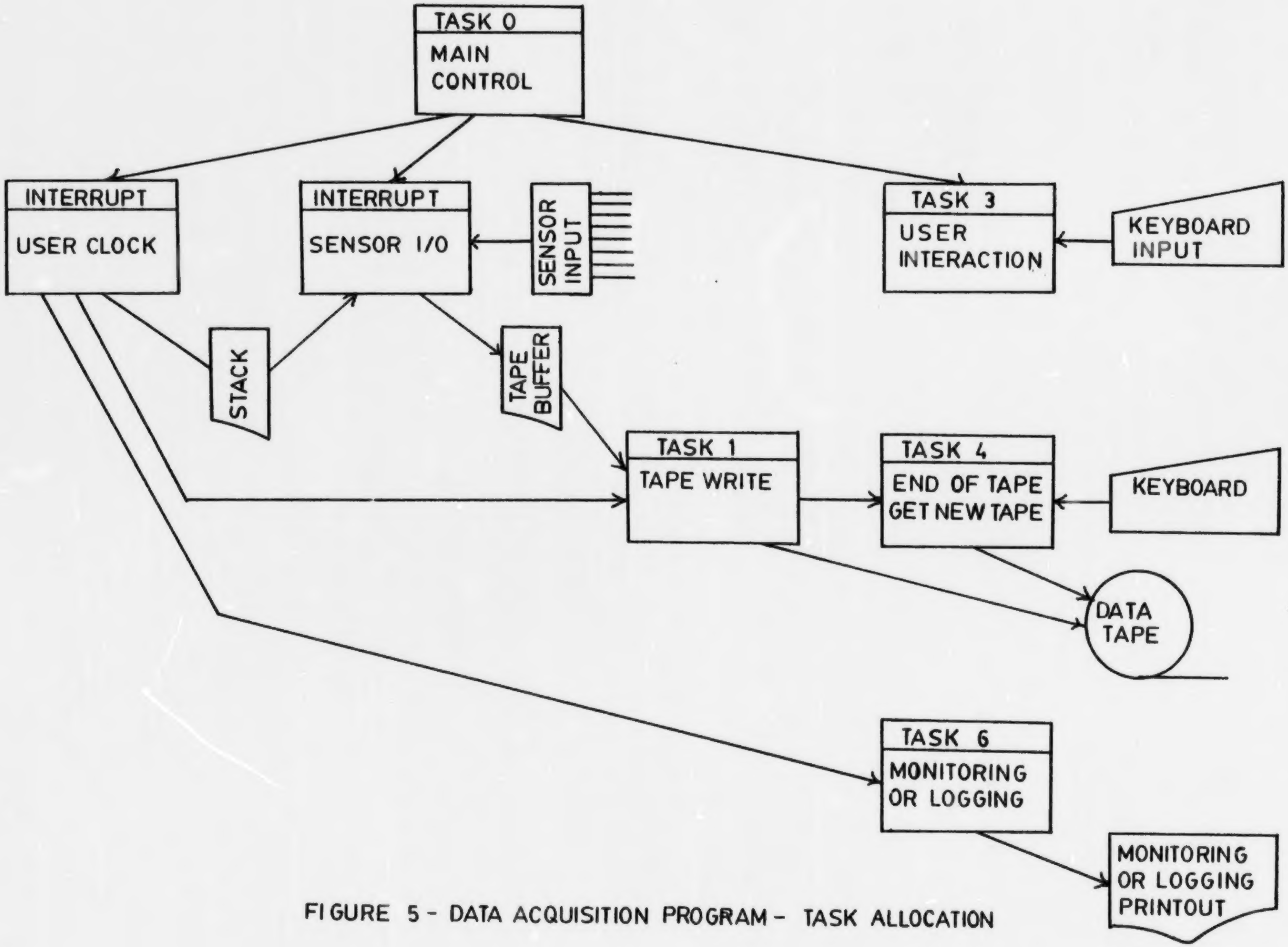
TASK 0
MAIN CONTROL
INTERRUPT
USER CLOCK
INTERRUPT
SENSOR I/O
SENSOR INPUT
TASK 3
USER INTERACTION
KEYBOARD INPUT
STACK
TAPE BUFFER
TASK 1
TAPE WRITE
TASK 4
END OF TAPE GET NEW TAPE
KEYBOARD
DATA TAPE
TASK 6
MONITORING OR LOGGING
MONITORING OR LOGGING PRINTOUT

FIGURE 5 - DATA ACQUISITION PROGRAM - TASK ALLOCATION

## Task Structure

| ID/PRIORITY | MODULE | COMMENTS |
|---|---|---|
| 0/0 | MV000 | Initialize |
| 0/2 | MV040 | Wait for ESC key |
| 1/1 | TR200 | Tape write task |
| 3/3 | MVTK3 | Operator input/intervention |
| 4/4 | TR400 | (End of old tape)/request new tape |
| 6/6 | MN500<br>LG500 | Monitoring or logging printout |

## Tasks

| ID | Pri | Title | Function |
|---|---|---|---|
| 0 | 0 | MV100 | Initial Setup |
| 0 | 0 | MV140 | Wait for ESC char/kill other type tasks |
| 0 | 3 | MV150 | Decide to stop recording or not |
| 0 | | MV160 | Kill recording |
| 0 | | MV165 | Keep on recording |
| 0 | | MV187 | Tape Request |
| 0 | | | a) Writing in progress |
| | | | b) No writing |
| 2 | 2 | TR200 | Tape write |
| | | TR240 | EOT processing |
| | | | a) Another tape ready |
| | | | b) Only tape |
| | Only used if two tapes, when swapping | | |
| 4 | 4 | TR300 | Finish old tape: EOF, release/Go on to TR400 |
| 4 | 4 | TR400 | Request a new alt tape |
| 6 | 6 | | Logging or monitoring printout (waits on . REC unless neither selected) |

### Task Combinations

```
0  (Wait)       2  (Write)          { 3 (intervention)
                                    { 4 (request)
                                    { 6 (logging/monitoring)

        A.  EOT encountered in task 2

            a)  Alternate not read

                EOF, release old tape (whole second)

              Kill 3, 4, or 6 (record which)/type signal     sub 300

                Request new tape/get/header
                If one-drive system

                    If  3 killed, ready  0
                    If  4 killed, create 6                   sub 350
                    If  6 killed, create 6

                    Start recording on even second
                If two drive system
                    Start task 4 - TR400

                        If  3 killed, ready  0
                        If  4 killed, ready  6               sub 350
                        If  6 killed, create 6

                        .KILL

                    Start recording on an even second
                End if

            b)  Alternate ready
                Swap channels (whole second)
                Write header or new

              Kill 3, 4, or 6 (record which)/type signal     sub 300

                Start task 4 - TR300
                    EOF, release old tape
                    Request new tape/get or not

                    If  3 killed, ready  0
                    If  4 killed, create 6                   sub 350
                    If  6 killed, create 6

                    .KILL
                  Start recording on even second
```

Task Combinations (Cont'd)

0 (Wait)  2 (Write)  6 (Writing or .REC wait)  0/0 2/2 6/6

A. EOT

a) One tape system

( EOF, release old tape (after full second)
( .KILL 6
( Suspend 0 or set MVINI = 1  suspend user clock?

Request new tape/get/write header
Wait for a start-second record (not half second)
Ready 0
Create 6
Resume writing on tape

b) Two tape system - second tape ready

Flip units
Write header
Wait for a start-second record
Resume writing on tape
Task end old, get new 4

( EOF, release old tape
( .KILL 6
( Suspend 0
Request new tape/get or not
Ready 0
Create 6
.KILL 4

c) Two tape system - second tape not ready

( .KILL 6
( Suspend 0
( EOF, Release old tape (after full record)

Request new tape/get/write header
Wait for a start-second record (not half word)
Resume writing
Task get alternate

Request new tape/get or not
Ready 0
Create 6

Task Combinations (Cont'd)

```
0 (Wait      2 (Write)     6 (Writing or .REC wait)    0/0   2/2   6/6

B. ESC char

   KILL 6                                              0/0   2/2         *
   Sept priority to 3                                  0/3   2/2         *
   Check modify or not

   a)  Modify sampling pattern

          Check to be sure
          Kill user clock )    not necessary
          Stop interface  )    safe, buth throws off
          EOF - tape           logging time count
          .KILL 2                                      0/3               *
          Set MVINI to modify
          Get & do all but tape
          On tape request
             EOF, release (?  Possibly not unless requested)
             TR100 as sub
          When no more mods:
          Write header
          Start Clock
          Create tape write task (2)                   0/3   2/2
        ( Create 6                                     0/3   2/2   6/6
        ( Set priority to 0                            0/0   2/2   6/6   *
        ( Revert to waiting

   b)  No mods                                         0/3   2/2
          Set MVINI to no mod
          Get & do all but tape
          On tape request:                             0/3
             Suspend 0                                  5    2/2
             TR100 as                                   5
               Ready 0                                 0/3
                                                       0/3
          When no more mods                            0/3   2/2
        ( Create 6                                     0/3   2/2   6/6
        ( Set priority to 0                            0/0   2/2   6/6
        ( Revert to waiting
```

### 2.4 Stack Structure

A stack of pointers controls sensor sampling. After the sampling pattern is selected, the addresses are sorted by rate into rate tables. At appropriate time intervals, the user clock interrupt service routine places pointers to these rate tables onto a LIFO stack. All sensors in the first table are sampled before the table pointer is popped off the stack and the next rate table used. However, new pointers are added to the stack at intervals. Therefore, it may happen that before all the sensors in one table have been sampled, a new table pointer is added to the stack. Then all the sensors in the new table are sampled before sampling of sensors in the original table is resumed.

Examples on the following two pages illustrate the working of the stack.

Stack Structure, Example 1

| Rate | Table Pointer | Sensors In Table |
|---|---|---|
| 250/sec | R6 | 1,2,3,4 |
| 100/sec | R5 | 5,6,7,8,9 |
| 40/sec | R4 | 10,11,12,13 |

| Milli-second | Sensors Sampled | Pointers added to Stack | Comments |
|---|---|---|---|
| | | R6,R5,R4 | |
| 1 | 1 2 3 | | R6 started |
| 2 | 4 5 6 | | R6 finished, R5 started |
| 3 | 7 8 9 | | R5 finished |
| 4 | 10 11 12 | R6 | R4 started/R6 stacked on partial R4 |
| 5 | 1 2 3 | | R6 started |
| 6 | 4 13 14 | | R6 finished/R4 finished |
| 7 | | | Stack emptied - no sampling |
| 8 | | R6 | No sampling/R6 pointer stacked |
| 9 | 1 2 3 | | R6 started |
| 10 | 4 | R5 | R6 finished/stack emptied/R5 pointer stacked |
| 11 | 5 6 7 | | R5 started |
| 12 | 8 9 | R6 | R5 finished/stack emptied/R6 pointer stacked |
| 13 | 1 2 3 | | R6 started |
| 14 | 4 | | R6 finished/stack emptied |
| 15 | | | No sampling |
| 16 | | R6 | No sampling/R6 pointer stacked |
| 17 | 1 2 3 | | R6 started |
| 18 | 4 | | R6 finished/stack emptied |
| 19 | | | No sampling |
| 20 | | R6,R5 | R6,R5 stacked |
| 21 | 1 2 3 | | R6 started |
| 22 | 4 5 6 | | R6 finished/R5 started |
| 23 | 7 8 9 | | R5 finished |
| 24 | | R6 | Stack empty/R6 stacked |
| 25 | 1 2 3 | R4 | R6 started/R4 stacked on partial R6 |
| 26 | 10 11 12 | | R4 started |
| 27 | 13 14 4 | | R4 finished/R6 finished |
| 28 | | R6 | No sampling/R6 stacked |

Stack Structure, Example 2

| Rate | Table Pointer | Sensors In Table |
|---|---|---|
| 1000/sec | R7 | 1,2 |
| 250/sec | R6 | 3,4 |
| 100/sec | R5 | 5,6 |
| 40/sec | R4 | 7,8,9 |

| Milli-second | Sensors Sampled | Pointers added to Stack | Comments |
|---|---|---|---|
| | | R7,R6,R5,R4 | |
| 1 | 1 2 3 | R7 | R6 started |
| 2 | 1 2 4 | R7 | |
| 3 | 1 2 5 | R7 | R5 started |
| 4 | 1 2 6 | R7,R6 | R6 stacked on top of R4 |
| 5 | 1 2 3 | R7 | New R6 started |
| 6 | 1 2 4 | R7 | |
| 7 | 1 2 7 | R7 | R4 started |
| 8 | 1 2 8 | R7,R6 | R6 stacked on top of partially used R4 |
| 9 | 1 2 3 | R7 | New R6 started |
| 10 | 1 2 4 | R7,R5 | |
| 11 | 1 2 5 | R7 | New R5 started |
| 12 | 1 2 6 | R7,R6 | |
| 13 | 1 2 3 | R7 | New R6 started |
| 14 | 1 2 4 | R7 | |
| 15 | 1 2 9 | R7 | Last of R4 table used |
| 16 | 1 2 | R7,R6 | Stack empties |
| 17 | 1 2 3 | R7 | New R6 started |
| 18 | 1 2 4 | R7 | |
| 19 | 1 2 | R7 | Stack empties |
| 20 | 1 2 | R7,R6,R5 | Stack empties |
| 21 | 1 2 3 | R7 | New R6 started |
| 22 | 1 2 4 | R7 - | |
| 23 | 1 2 5 | R7 | New R5 started |
| 24 | 1 2 6 | R7,R6 | |
| 25 | 1 2 3 | R7,R4 | R4 stacked on top of partial R6 |
| 26 | 1 2 7 | R7 | R4 sensors sampled first |
| 27 | 1 2 8 | R7 | |
| 28 | 1 2 9 | R7,R6 | R6 stacked on top of partial R6 |
| 29 | 1 2 3 | R7 | |
| 30 | 1 2 4 | R7,R5 | R5 stacked on top of partial R6 |

Stack Structure, Example 2 (Cont'd)

| Millisecond | Sensors Sampled | Pointers added to Stack | Comments | Rate | Table Pointer | Sensors In Table |
|---|---|---|---|---|---|---|
| | | | | 1000/sec | R7 | 1,2 |
| | | R7,R6,R5,R4 | | 250/sec | R6 | 3,4 |
| | | | | 100/sec | R5 | 5,6 |
| | | | | 40/sec | R4 | 7,8,9 |
| 31 | 1 2 5 | R7 | | | | |
| 32 | 1 2 6 | R7,R6 | R6 stacked on top of partial R6 | | | |
| 33 | 1 2 3 | R7 | | | | |
| 34 | 1 2 4 | R7 | | | | |
| 35 | 1 2 4 | R7 | Partial R6 finally used | | | |

## 2.5 Utility Subroutines

- OC.AS - Binary to ASCII octal - 6 characters
  - AC0 - byte pointer, updated
  - AC1 - number to be converted
- BN.AS - Binary to ASCII decimal - sign +5 characters
  - AC0 - byte pointer, updated
  - AC1 - number to be converted
- DIG. - ASCII decimal to binary
  - AC0 - byte pointer, updated
  - AC1 - answer output
  - AC2 - # digits output
- DIG.N - ASCII decimal to binary - N digits
  - AC0 - byte pointer, updated
  - AC1 - N (number of digits) input
    answer output
  - RETURN - error
  - RETURN+1 - OK
- GT.CH - Get next character
  - AC0 - byte pointer, updated
  - AC1 - character output
- R.LN - read a line into RDBF (IBUF = byte pointer)
  - RETURN - CR only
  - RETURN+1 - OK
- P.CR - Print a line w/pad character at start
  - AC0 - byte pointer
- P.NL - Print a line - no pad character
  - AC0 - byte pointer
- MUL.A
  - AC0,AC1 = AC1 x AC2 +AC0
- MUL.U
  - AC0,AC1 = AC1 x AC2
- DIV.I - AC1/AC2
  - AC0 - remainder
  - AC1 - answer
- DIV.U - AC0,AC1/AC2
  - AC0 - remainder
  - AC1 - answer

## Magnetic Tape Subroutine Summary

MT.AS - Assign tape drive: request unit, init, open
- AC0 - byte pointer to message (input)
- AC2 - pointer to MT Table (returned)
- RETURN - just CR
- RETURN+1 - successful

MT.DV - Decode & return # of tape drives (1 or 2)
- AC1 - return 1 or 2
- RETURN - error
- RETURN+1 - OK

MT.UN - Decode tape unit #, return MT Table pointer
- AC1 - return 0 to 1
- AC2 - return MT Table pointer
- RETURN - error (prints message w/no CR)
- RETURN+1 - OK

MT.IN - Initialize & open - does *not* activate channel
- AC2 - MT Table pointer (input)

MT.HD - Build & write a header
- AC2 - MT Table pointer (input)

MT.EF - Write two EOF's
- AC2 - MT Table pointer (input)

MT.RL - Close, release, deactivate
- AC2 - MT Table pointer (input)

## Utility Subroutines

A. BN.AS - Binary to ASCII (decimal) conversion +/- MV2/BNASC

1. Input: AC0 - byte pointer where ASCII is to go
   AC1 - number to be converted

2. Output: AC0 - updated byte pointer
   Print area - 6ASCII characters, digits

B. OC.AS - Binary to ASCII (octal) conversion 0CASC

1. Input: AC0 - byte pointer where ASCII is to go
   AC1 - number to be converted

2. Output: AC0 - updated byte pointer
   Print area - 6 ASCII digits

C. DIG - ASCII (decimal) to binary conversion DIG

1. Input: AC0 - byte pointer
2. Output: AC0 - updated byte pointer (points to non-digit)
   AC1 - number converted (answer)
   AC2 - number of digits found

D. DIG.N - ASCII (decimal) to binary conversion of N digit DIGN

1. Input: AC0 - byte pointer
   AC1 - number of digits to be converted

2. Output: AC0 - byte pointer
   AC1 - number converted (answer)

3. Return: Normal - error: non-digit found
   Normal +1 - conversion OK

E. GT.CH - Get one character from ASCII string GTCH

1. Input: AC0 - byte pointer
   AC1 - character returned in right byte

F. R.LN - Read a line from the terminal RLN
1. Input: None
2. Output: Line in input buffer
3. Return: Normal - one character (just CR) read
Normal +1 - line read

G. P.CR - Print a line (at start of line) w/pad character PCR
1. Input: AC0 - byte pointer
2. Output: Printed line

H. P.NL - Print a line (continuation) w/o pad character PNL
1. Input: AC0 - byte pointer
2. Output: Printed line

I. MUL.U - Unsigned integer multiply AC0,AC1←AC1xAC2 MV2/MULU
1. Input: AC1 - multiplicand
AC2 - multiplier
2. Output: AC0 - high order part of answer
AC1 - low order part of answer

J. MUL.A - Unsigned integer multiply and add AC0,AC1←AC2+AC0 MULA
1. Input: AC0 - adder
AC1 - multiplicand
AC2 - multiplier
2. Output: AC0 - high order part of answer
AC1 - low order part of answer

K. DIV.I - Single precision divide DIVI
1. Input: AC1 - dividend
AC2 - divisor
2. Output: AC0 - remainder
AC1 - answer

L. DIV.U - Double precision divide DIVU
1. Input: ACO - high order part of dividend
AC1 - low order part of dividend
AC2 - divisor

2. Output: ACO - remainder
AC1 - answer

M. C.VLT - Binary to ASCII (volts, +/-10.00) conversion CVLT
1. Input: ACO - even byte pointer
AC1 - value (0 to 1023)

2. Output: ACO - not updated
Print area - 3 words -xx.xx format

N. G.ADR - Get ADR or ADR-ADR or ADR-ADR/INC MV4/GADR
1. Input: ACO - byte pointer
2. Output: ACO - updated byte pointer
LADRI - first adr
LADR2 - second adr if given, else first adr
LINCR - increment if given, else L
ERCT - increment if

3. Return: Normal - done, status reported
- error, comma in check only mode
Normal + 1 - go on, comma in modify mode

O. R.PS - report present status of ADR (i.e., R,M,MIN,MAS) MV4 RPS
1. Input: ADR - address to be reported
TABJI - address of entry in TAB1

2. Output: Printed line

P. WRN.AB - warning, chance to abort WRNAB
1. Input: ACO - byte pointer to specific message
User response: C or otherwise

2. Output: Specific message + "TYPE C TO CONTINUE"

3. Return: Normal - No C, abort
Normal + 1 - C, continue

Q. P.ERR - Print error counts at end of tape file PERR

1. Input: SPURZ - count of stack pointer zeroes
   TPOVF - count of tape buffer overflows

2. Output: SPURZ - zeroed
   TPOVF - zeroes
   Counts printed if non-zero

R. MT.AS - Assign a mag tape MVMT/MTASG

1. Input: AC0 - byte pointer to specific message
   User enters unit number

2. Processing: Get unit number from user
   Check unit unused
   Initialize and open

3. Output & Return:
   a) If unit, selected
      AC2 pointer to selected MT Table
      Normal return +1
   b) If no unit selected (just CR)
      AC2 indeterminate
      Normal return

S. MT.DV - decode, & check # of MT units (1 or 2) MTDV

1. Input: Line in input buffer

2. Output: AC1 - returns 1 or 2

3. Return: Normal - error: neither 1 or 2
   Normal +1 - OK

T. MT.UN - Decode and check MT unit # (0 or 1) MTUN

1. Input: line in input buffer

2. Output: AC1 - returns 0 or 1
   AC2 - returns pointer to selected MT Table

3. Return: Normal - error: neither 0 nor 1
   Normal +1 - OK

U. MT.IN - Initialize and open selected mag tape unit MTINI

1. Input: AC2 - pointer to selected MT Table
2. Output: two EOF's (full initialization)

V. MT.HD - Write header to selected mag tape unit MTHED

1. Input: AC2 - pointer to selected MT Table
2. Output: tape header

W. MT.EF - Write 2 EOF's to selected mag tape unit MTEF

1. Input: AC2 - pointer to selected MT Table
2. Output: two EOF's (end of tape)

X. MT.RL - close, rebase and deactivate MT unit MTREL

1. Input: AC2 - pointer to selected MT Table
2. Output: tape rewound

## 2.6 Magnetic Tape Formats

Data tapes are written at 800 bpi in NZRI format with odd parity (IBM compatible). Nine-track tapes are used. A data tape contains one or more data files. A data file consists of a header block, a variable number of data blocks, and an end-of-file mark. Two consecutive end-of-file marks denote the end of tape (no more data files recorded on the tape.)

## Header Block Format

A header block is of fixed length, containing 1546 words of 16 bits each. A header block occurs at the beginning of every file. The conditions which cause one file to be ended and a new one begun are:

1) End of tape - when one tape is full of data, recording on a new tape begins with a header block.

2) When the operator halts data gathering, the current file is ended. When data gathering is resumed, the new file begins with a header block.

| Word | Contents | |
|---|---|---|
| 1 | Year | Date & Time (6 words) |
| 2 | Month | |
| 3 | Day | |
| 4 | Hour | |
| 5 | Minute | |
| 6 | Second | |
| 7 | Logging interval (in minutes) | |
| 8 | Recording mode (-1 for recording; 0 for standby) | |
| 9 | Number of sensors (256 or 512) | |
| 10 | - spare - | |
| 11-522 | Rate and mode table (512 words) | |

| Bits | Contents (value) |
|---|---|
| (msb) 0-2 | Unused (0) |
| 3-5 | Rate (0-7) |
| 6 | Mode (0=P, 1=R) |
| 7-15 (lsb) | Address (1-255) |

| Word | Contents |
|---|---|
| 523-1034 | Alarm limit table - minimums (512 words) |
| 1035-1546 | Alarm limit table - maximums (512 words) |

| Word | Contents | |
|---|---|---|
| 1 | Year | start date & time<br>6 words |
| 2 | Month | |
| 3 | Day | |
| 4 | Hour | |
| 5 | Minute | |
| 6 | Second | |
| 7 | Logging interval | |
| 8 | Recording mode | |
| 9 | Number of sensors (256 or 512) | |
| 10 | - spare - | |
| 11 | default rate, default mode, adr 0 | Sampling pattern table<br>512 words |
| | rate$_1$, mode$_1$, adr 1 | |
| | rate$_2$, mode$_2$, adr 2 | |
| | : | |
| | . | |
| | rate$_{255}$, mode$_{255}$, adr 255 | |
| | space for adr 256 | |
| | : | |
| | . | |
| 522 | Space for adr 511 | |
| 523 | default min | Minimum alarm limit<br>table 512 words |
| | min$_1$ | |
| | min$_2$ | |
| | : | |
| | . | |
| | min$_{255}$ | |
| | min$_{256}$ | |
| | : | |
| | . | |
| 1034 | Space for min$_{511}$ | |

Header Block (Cont'd)

| Word | Contents | |
|---|---|---|
| 1035 | default max | Maximum alarm limit table 512 words |
| | $\text{max}_1$ | |
| | $\text{max}_2$ | |
| | : | |
| | . | |
| | $\text{max}_{255}$ | |
| | Space for max 256 | |
| | : | |
| | . | |
| 1546 | Space for max 511 | |

## Data Block Format

When the system is recording, two data blocks are recorded each second. A data block is of variable length, from a minimum of 10 words, to a maximum of 4096 words. The first eight words specify the date and time of the record, the recording mode, and whether this is the first or second record of the pair for the second. The following words are data records, grouped in pairs. The first word in each pair specifies the address of the sensor sampled (1 to 255). The second word is the reading obtained from the sensor (0 to 1023).

| Word | | |
|---|---|---|
| 1-6 | Date and time (same as header) | |
| 7 | Block type | |
| | -1 Continuous recording | |
| | n Standby triggered into recording (n more blocks will be recorded) N is decremented in successive records. | |
| | 1 Standby logging (once per hour two successive records with 1 in this field are written) | |
| 8 | Index number | |
| 9 | Sensor address (1-255) | Data record |
| 10 | Reading (0-1023; 0= -10V, 1023= +10V | |
| 11-12 | Data Record | |
| : | : | |

DATA BLOCK

| Word | | |
|---|---|---|
| 1 | Year | date & time of this record<br>6 words |
| 2 | Month | |
| 3 | Day | |
| 4 | Hour | |
| 5 | Minute | |
| 6 | Second | |
| 7 | Block type (continuous, standby, logging) | |
| 8 | Index (1st or 2nd record for this send) | |
| 9 | Adr$_1$ | data record<br>2 words |
| 10 | Data reading$_1$ | |
| 11 | Adr$_2$ | data record<br>2 words |
| 12 | Data reading$_2$ | |
| | : | |
| | . | |
| 8+2n-1 | Adr$_n$ | data record<br>2 words |
| 8+2n | Data reading$_n$ | |

## 2.7 Program Structure Outline

I. Overview

| | | | |
|---|---|---|---|
| A. | Initialization | MVO | MV000 |
| | 1. I/O initialization | MVO | |
| | 2. One-time initialization | MVO | |
| | 3. Allow modification of sampling pattern and alarm limits | | |
| B. | User interaction - parameters initialization | MVO | |
| | 1. Prompting mode | MV8 | VB000 |
| | 2. Date & time | MVDT | DT000 |
| | 3. Sampling pattern | MVSP | SP000 |
| | 4. Alarm limits | MVAL | AL000 |
| | 5. Logging | MVLG | LG000 |
| | 6. Monitoring | MVMN | MN000 |
| | 7. Recording mode | MVRM | RM000 |
| | 8. Tape unit initialization | MVTR | TR000 |
| C. | Option to modify or continue | MVO | |
| D. | Start new data gathering | MVO | MV040 |
| | 1. Write tape header | MVMT | MTHED |
| | 2. Start user clock | MV3 | UCLST |
| | 3. Create tape write task | MVTR | TR200 |
| E. | Continue data gathering | MVO | MV050 |
| | 1. Start monitoring | MVMN | MN500 |
| | 2. If no monitoring adr, start logging | MVLG | LG500 |
| F. | Wait for operator intervention | MVO | |
| | 1. Set priority to 0 (highest) | MVO | |
| | 2. Wait for ESC character | MVO | |
| G. | Do operator intervention | MVO | |
| | 1. Decrease priority to 3 (less than tape write) | MVO | |
| | 2. Kill monitoring, logging, or tape request | MVO | |
| | 3. Create user intervention task | MVO | MV100 |
| | 4. Suspend self/restart at D or E | MVO | |

| | | |
|---|---|---|
| II. Utility Routines | | |
| A. General subroutines | MV2 | |
| B. Special subroutines | MV4 | |
| C. Mag tape subroutines | MVMT | |
| III. I/O and interrupt processing | MV3 | |
| A. User clock | MV3 | |
| 1. Start routine | MV3 | UCLST |
| 2. Interrupt service routine | MV3 | UCLIS |
| B. Stack push routine | MV3 | SPUSH |
| C. Interface start routine | MV3 | INST |
| D. Interface interrupt service routine | MV3 | |
| 1. Input device | MV3 | INIS |
| 2. Output device | OUTIS | |
| IV Test routines | MV1 | |
| A. Basic hardware test | MV1 | TEST |
| B. Buffered hardware test | MV1 | MV200 |
| C. Mag tape test | MV1 | TT000 |

```
I-B-1 Prompting Mode                                       MV8

Type welcome                                               MV8     VB000
Ask if verbose prompting desired/set flag                  MV8

Processing Common to all Subsequent Entries                MV8     VB900

If minimal prompting,
    Return
Else
    Get pointers & count for appropriate message           MV8
    Type message                                           MV8
    Return                                                 MV8
End if

I-B-2 Date & Time Selection                                MVDT    DT000

Optional verbose prompting                                 MV8     VB100
If date has once been set                                  MVDT
    Type present date & time                               MVDT    DT100
End if                                                     MVDT
Ask for date                                               MVDT    DT010
If date entered, set date                                  MVDT

Ask for time                                               MVDT    DT020
If time entered, set time                                  MVDT

If changed, print current date & time                      MVDT    DT100
Return                                                     MVDT

I-B-3 Sampling Pattern Selection                           MVSP    SP000

Optional verbose prompting                                 MV8     VB200
If modify permitted                                        MVSP
    Ask INIT, MOD, or CHECK?                               MVSP
    Set IFMOD for INIT, MOD ; Reset for check              MVSP
```

## I-B-3 Sampling Pattern Selection (Cont'd)

```
Else
      Reset IFMOD                                        MVSP    SP015
End if
If INIT                                                  MVSP
      Warning, chance to abort                           MV4     WRNAB
      Initialize TAB1 (not min/max tables)               MVSP
End if
Type appropriate prompt                                  MVSP
Type # samples/sec to go                                 MVSP    SP450

Loop to report or modify                                 MVSP    SP100

Issue prompt
If input received
      Decode ADR or ADR-ADR or ADR-ADR/INCR              MV4     GADR
      If that's all                                      MV4     GADR
          Print the requested status                     MV4     GADR
          Return done - go to prompt                     MV4     GADR
      If there's more and this is check mate             MV4     GADR
          Print error message                            MV4     GADR
          Return error - go to prompt                    MV4     GADR
      If there's more & mods are OK                      MVSP
          Get, check & save rate                         MVSP
          Get, check & save mode                         MVSP
          If any errors, go to prompt                    MVSP    SP160
          Else                                           MVSP
              If zero rate                               MVSP
                    Check, may delete monitoring         MVSP
                    Check, may delete Alarm B            MVSP
              End if                                     MVSP
              Loop to modify TAB1                        MVSP    SP170
              Check samples/sec (prospective)            MVSP    SP400
              If OK                                      MVSP
                    Modify TAB1                          MVSP
                    Loop
```

## I-B-3 Sampling Pattern Selection (Cont'd)

```
          In no good                                     MVSP    SP180
              Type offending ADR                         MVSP
              end loop                                   MVSP
          End if                                         MVSP
          At ENDLOOP                                     MVSP
              Type samples/sec to go                     MVSP    SP190
              Go issue prompt                            MVSP    SP450
          End if                                         MVSP
      End if                                             MVSP
If just CR typed                                         MVSP    SP800
      Sort TAB1 into up to 7 tables by rate              MVSP
      Follow last adr in each table by zero              MVSP
      Exit SAMPLING PATTERN SELECTION                    MVSP    SP900
End if

I-B-4 Alarm Limit Selection                              MVAL    AL200

Optional verbose prompting                               MVB     VB300
If modify permitted                                      MVAL    AL010
      ASK INIT, MOD or CK?                               MVAL
      Set IFMOD  for INIT, MOD; reset for check
Else
      Reset IFMOD                                        MVAL    AL015
End if
If INIT                                                  MVAL    AL020
      Warning, chance to abort                           MV4     WRNAB
      Initialize TAB3 (minima) & TAB4 (maxima)           MVAL    AL040
End if
Type appropriate prompt                                  MVAL    AL090

Loop to report or modify

Issue prompt                                             MVAL    AL100
In input received
      Decode ADR or ADR-ADR or ADR-ADR/INCR              MV4     GADR
      If that's all
          Print the requested status                     MV4     GADR
```

## I-B-4 Alarm Limit Selection (Cont'd)

```
            Return done - go to prompt                      MV4     GADR
        If there's more and this is check mode
            Print error message                             MV4     GADR
            Return error - go to prompt                     MV4     GADR
        If there's more and mods are OK                     MVAL    AL100
            Get, check and save min (in none, use existing) MVAL    AL100
            Get, check and save max (if none, use existing) MVAL
            If any errors, go to prompt                     MVAL    AL160
            Else loop to modify TAB3,TAB4                   MVAL    AL170
        End if
    If just CR typed                                        MVAL
        Exit ALARM LIMIT SELECTION                          MVAL    AL900
    End if
    I-B-5 Logging Interval Selection                        MVLG    LG000
    Optional verbose prompting                              MV8     VB400
    Type present interval                                   MVLG    LG100
    Ask for new interval                                    MVLG    LG015
    If data input                                           MVLG
        Check value                                         MVLG
        If zero,  zero minutes, hours, minute count         MVLG    LG070
        Else
        Check units                                         MVLG    LG030
        If units not specified                              MVLG
            Ask HOURS OR MINUTES                            MVLG
            Get and check input                             MVLG
        End if
        If hours,                                           MVLG    LG040
            Save in hours                                   MVLG
            Multiply by 60 for minutes                      MVLG
        If minutes                                          MVLG    LG050
            Save in minutes                                 MVLG
            zero hours                                      MVLG
        End if
        Start minute count at 1                             MVLG    LG060
```

## I-B-5 Logging Interval Selection (Cont'd)

```
     Type present interval                               MVLG   LG100
End if
Exit LOGGING INTERVAL SELECTION                          MVLG   LG090
```

## I-B-6 Monitoring Selection

```
                                                         MVMN   MN000
Optional verbose prompting - MONITORING                  MV8    VB500
Type present status                                      MVMN   MN100
Ask for new ADR                                          MVMN   MN420
If data input                                            MVMN
     Check adr                                           MVMN
     If not sampled, error message & try again           MVMN   MN240
     Else
          Save adr                                       MVMN   MN445
          Type present status                            MVMN   MN100
     End if
End if                                                   MVMN
Optional verbose prompting - ALARM                       MV8    VB600
Type present status                                      MVMN   MN100
Ask for new ADR                                          MVMN   MN420
If data input                                            MVMN
     Check adr                                           MVMN
     If not sampled                                      MVMN
          Error message & try again                      MVMN   MN240
     Else
          Save adr                                       MVMN   MN445
          Type present status                            MVMN   MN100
     End if                                              MVMN
End if                                                   MVMN
Exit MONITORING SELECTION                                MVMN
```

```
I-B-7 Recording Mode Selection                          MVRM    RM000

Optional verbose prompting                              MV8     VB700
Type present status                                     MVRM    RM100
Ask for new selection                                   MVRM    RM010
If data input                                           MVRM
      If standby selected                               MVRM
           Ask for new duration                         MVRM    RM020
           If data input                                MVRM
               Store response x 2 for tape-write count  MVRM
           End if                                       MVRM
      End if                                            MVRM
      Store indication                                  MVRM    RM030
      Type present status                               MVRM    RM100
End if                                                  MVRM
Exit RECORDING MODE SELECTION                           MVRM    RM090

I-B-8 Initial Mag Tape Selection                        MVTR    TR000

Optional verbose prompting                              MV8     VB800
Zero MT Table pointers                                  MVTR    TR000
Get # devices (1 or 2) & save                           MVTR    TR010
Ask for unit # for writing                              MVTR    TR160
Activate channel                                        MVTR
If two devices                                          MVTR    TR170
      Ask for unit # for alternate                      MVTR
      If device entered                                 MVTR
           Activate channel                             MVTR
      Else
           Don't                                        MVTR
      End if                                            MVTR
End if                                                  MVTR
Exit INITIAL MAG TAPE SELECTION                         MVTR    TR190
```

## 2.8 Conversion Equations

The conversion of the output of the ADC in binary units to engineering units is performed within the program PVAL. The program receives the sensor address (1 to 255) in AC2 and the sampled value N(0-1023) in AC1. The sensor address is used to index into STAB which contains a one-word pointer to the binary-to-engineering units conversion block for each address. The block contains the address of the conversion subroutine, as well as the necessary constants, and pointers to the literals for units.

A block of code is set up as follows:

| | | |
|---|---|---|
| Line 1 | NAME: | byte pointer to label for units |
| 2 | | byte pointer to label for units if constant,<br>0 if units change with sign |
| 3 | | subroutine name |
| 4 | | scale factor |
| 5 | | A - highest order coefficient |
| 6 | | B - second highest order coefficient |
| 7 | | byte pointer to units for positive value |
| 8 | | byte pointer to units for negative value |
| 9 | | C - third highest order coefficient |
| 10 | | D - fourth highest order coefficient |
| 11 | LABEL 1: | literals of units when units do not change with sign |
| 12 | LABEL 2: | literals of units for positive value |
| 13 | LABEL 3: | literals of units for negative value |

Line 1 NAME is the label for that block of code. The address of this label is obtained by the main program by indexing through STAB.

The byte pointer indicates the label at which location the units (literals) are to be found.

Line 2 When the units do not change with sign, lines 1 and 2 are identical. When the units change with sign, line 2 is 0.

Line 3 The subroutine name indicates the form of the conversion equation (linear, quadratic, etc.). All conversion equations are functions of (N/1024), where N is the sampled value of that sensor (value between 0 through 1023).

| Subroutine | Equation | Type |
|---|---|---|
| LINP | $Ax$ | Linear in x, no constant |
| LINC | $Ax + B$ | Linear in x, with constant units do not change with sign |
| LINU | $Ax + B$ | Linear in x, with constant units change with sign |
| POLY2 | $Ax^2 + Bx + C$ | Quadratic in x |
| POLY3 | $Ax^3 + Bx^2 + Cx + D$ | Trinomial in x |

In all the above equations $x = N/1024$. All coefficients are entered as integers. Positioning of the decimal point is determined by the choice of the scale factor.

| | |
|---|---|
| Line 4 | The scale factor is a number between 1 through 4. When the scale factor chosen is s, then the decimal point is placed before the (4-s) least significant digits in each coefficient. When the highest order coefficient A is negative, reverse the sign of all coefficients and add 1B0 to the scale factor. |
| Line 5 | A is the highest order coefficient according to the conversion equation and must always be positive. |
| Line 6 | B is the second highest order coefficient; it may be omitted when zero. |
| Lines 7, 8 | These are byte pointers to the units for positive and negative values respectively. They may be omitted where units remain unchanged. |
| Lines 9, 10 | C and D are the third and fourth highest order coefficients respectively. They may be omitted when zero. (C only when D is also zero.) |
| Lines 11, 12, 13 | These contain literals to print units. The lines may or may not be located directly after the code or may be located elsewhere in the program. |

## Output in Volts

When the printout is required in volts instead of engineering units (during calibration, etc.) the contents of label IFVLT (by default 0) should be changed to 1. This modification can be carried out either through the Symbolic Debugger (RTDEB03) or with the Octal Editor. When IFVLT = 1 all sensors use the following conversion equation:

$$\text{Output in Volts} = 20\,(N/1024) - 10 \qquad \text{(i)}$$

## Examples of Code Blocks

i) Named SVOLT, converts binary value (0 through 1023) to volts (-10V to +10V) according to above equation (i)

```
SVOLT:   TE001*2
         TE001*2
         LINC
         2
         2000
         -1000


TE001:   .TXT   /VOLTS <15>/
```

ii) Named STEST, converts binary value 0 through 1023 to degrees, (positive forward, negative reverse) according to the equation

$$y = -\,3.28x^3 + 0.79x^2 - 11.3\,x + 1$$

```
STEST:   UNITS 1*2

         POLY3
         2 + 180
         328
         -79.
         UNITS 2*2
         UNITS 3*2
         1130
         -100


UNITS 1: .TXT  /DEGREES      <15>/

UNITS 2: .TXT  /DEGREES STBD <15>/

UNITS 3: .TXT  /DEGREES PORT <15>/
```

## 2.9 Creating a Save File MV.SV

The following procedure should be followed in order to create a save file of the Data Acquisition program, ready for execution under RTOS.

### i) MACRO Assembly of Source Files

The MACRO Assembler translates symbolic language (a source program) into machine language (an object program). The assembler simply substitutes numeric code for symbolic instruction codes and numeric addresses (either absolute or relocatable as defined) for symbolic addresses.

The Data Acquisition Program Source files are assembled using the MAC command, and relocatable binary files obtained for each. Files to be assembled are:

MV0
MV1.
MV2.
MV3.
MV8.
MVPV.
MVVB.
MVSP.
MVAL.
MVLG.
MVMN.
MVTR.
MVMT.

These names are contained in file 'FILESMV'

MV<0, 1, 2, 3, 8, PV, VB, SP, AL, LG, MN, TR, MT>

### ii) RTOS System Generation [MV.TR]

This procedure will produce a relocatable binary which will trigger the loading of appropriate device drivers from the RTOS library, allocate tables and control blocks used by the system, and a fixed number of channels and task control blocks. Use of the system generation program RTOSGEN for this program is shown on the next page. The program is named MV.TR.

```
RTOSGEN


 RTOS  SYSTEM GENERATION REV 1.50

 CPU TYPE(03/02/64) ? 03
  CORE STORAGE (IN K WORDS) 32

  RTC FREQ (0=10HZ,1=100HZ,2=1000HZ,3=LINE) 2

  TASKS(1-255) ? 5
  CHANNELS(1-63) ? 4

RESPOND WITH NUMBER OF UNITS

  DSK(0-1) ? 0
  DKP(0-4) ? 0
  DFL(0-4) ? 0
  DSM(0-4) ? 0
  MTA(0-8) ? 2
    BUFFERED I/O (N/Y) ? N
  PTR(0-2) ? 0
  PTP(0-2) ? 0
  LPT(0-2) ? 0
  CDR(0-2) ? 0
  QTY(0-4) ? 0
  TTYS(0-2) ? 1
  MCA(0-15) ? 0
  AUTO RESTART (Y/N) ? Y


  HIGH PRIORITY INTERRUPTS(N/Y) ? N

  USER SUPPLIED DRIVERS , (N/Y) ? N

      SUMMARY OF RTOS SYSGEN

      CODE      EXTERNAL   NAME

        10      TTIDC      $TTI
        11      TTODC      $TTO
        14      RTCDC
        22      MTADC

  SYSGEN OK (Y/N) ? Y
  OUTPUT FILENAME ? MV.TR
R
```

iii) Get Libraries

The following libraries are required:

| | |
|---|---|
| MTADR.LB | Magnetic Tape Support |
| RTOS1.LB | Standard Libraries |
| RTOS2.LB | supporting general RTOS Functions |

If a debugger is to be included, use:

| | |
|---|---|
| RTDEB03.RB | Standard 1603 RTOS debugger |

iv) Relocatable Load

Using the Relocatable Loader (RLDR command) the relocatable binary files, libraries, and debugger are combined with the RTOSGEN system to create a save file ready for execution under RTOS. The command string for this operation is contained in file 'LDMV1'.

```
RLDR/C/D @FILESMV@ MV.TR <<MTADR, RTOS<1,2>>.LB> MV1.SV/S RTDEB03
```

The load map generated is shown on the next page.

When the debugger is not required use the command line:

```
RLDR/C  @FILESMV@  MV.TR  <<MTADR, RTOS<1,2>>.LB>  MV1.SV/S
```

```
@LDMU10
MU1.SV  LOADED BY RLDR REV 05.50 AT 13:47:55 10/31/80
   MU0   000440
   MU1   001255
   MU2   010410
   MU3   011130
   PVAL  011662
   MU8   014274
   RM100 015041
   VB000 015267
   DT000 024764
   SP000 025372
   AL000 026721
   LG000 027304
   MN000 027762
   TR000 030301
   MT000 031157
   RTOS  032072
   MTADR 032376
   MTSER 032456
   SMTU  033412
   REC   033442
   IXMT  033442
   TXMT  033442
   TIDR  033602
   TIDK  033602
   TIDC  033602
   PRI   033756
   TPRI  033756
   ABORT 034001
   TABT  034001
   TUMOD 034072
   TASK  034107
   KILL  034107
   TCBMO 034107
   NSAC3 034522
   DUMMY 034522
   INTD  034531
   TTYDR 034764
   RTCDR 035365
   PWRDR 036001
   SYSTE 036202
   BOOT  037234
   RTIN  037357
   SCHED 037633
   GENIO 037756
   IOSER 040531
   ABORT 041274
   RESOL 041427
   SADEB 041427


    NMAX 051373
    ZMAX 000341
    CSZE 000000
     EST 046270
     SST 051372

R
```

The disk directory of all software required to perform various maintenance operations related to the Data Acquisition Program is listed on the next page.

```
LIST/A/E/S
$LPT.                  0  RAP        01/01/68 00:00  01/01/68  [000000]     1
$PTR.                  0  APW        01/01/68 00:00  01/01/68  [000000]     1
$TTI.                  0  APW        01/01/68 00:00  01/01/68  [000000]     1
$TTO.                  0  RAP        01/01/68 00:00  01/01/68  [000000]     1
$TTP.                  0  RAP        01/01/68 00:00  01/01/68  [000000]     1
$TTR.                  0  APW        01/01/68 00:00  01/01/68  [000000]     1
ACTION.               40  D          10/31/80 06:19  10/31/80  [002157]     0
ACTION.BU             26  D          10/31/80 06:13  10/31/80  [002131]     0
BOOT.SV             6144  SPWD       06/21/78 21:16  04/25/79  [001724]     0
BOOTSYS.OL         32768  PC         10/25/80 04:30  10/25/80  [000020]     0
BOOTSYS.SV         46080  SPWD       12/21/77 00:22  04/25/79  [004436]     0
CLI.ER              4096  PWD        08/25/78 16:58  10/31/80  [005523]     1
CLI.OL             50176  PWC        08/25/78 16:58  10/31/80  [010006]     1
CLI.SV             14336  SPWD       08/25/78 16:58  04/25/79  [007763]     0
COM.CM               268             10/31/80 13:47  10/31/80  [000016]     0
EDIT.SV             7168  SPWD       11/22/77 11:24  04/25/79  [001133]     0
FILESMV.              44  D          10/31/80 13:46  10/31/80  [003312]     0
FILESMV.BU            39  D          10/31/80 12:23  10/31/80  [022211]     0
LDMV1.                66  D          10/22/80 19:38  10/31/80  [002227]     0
LDMV1.BU              66  D          02/14/80 15:53  10/22/80  [001176]     0
MAC.PS             16896  PWD        11/21/77 11:27  10/31/80  [011150]     0
MAC.ST             16896  D          10/31/80 21:02  10/31/80  [017722]     0
MAC.SV             18944  SPWD       11/21/77 11:26  04/25/79  [012226]     0
MAC.XR                 0  D          10/31/80 20:59  10/31/80  [001175]     0
MACXR.SV            3584  SPWD       11/21/77 11:27  04/25/79  [020056]     0
MAP.DR               152  APWC       10/25/80 04:30  10/25/80  [000017]     0
MTADR.LB            4164  PWD/PW     02/24/78 15:25  10/31/80  [001767]     0
MV.TR               1048             10/31/80 12:21  10/31/80  [003305]     0
MV0.                7693  D          10/11/80 21:16  10/31/80  [020001]     0
MV0.BU              7688  D          10/11/80 21:08  10/11/80  [013372]     0
MV0.RB              1686  D          10/11/80 21:22  10/31/80  [004426]     0
MV1.               10254  D          10/11/80 19:05  10/31/80  [002231]     0
MV1.BU             10254             05/27/80 08:20  10/11/80  [020031]     0
MV1.RB              2008  D          10/11/80 19:09  10/31/80  [012211]     0
MV1.SV             42496  SD         10/31/80 13:50  10/31/80  [003311]     0
MV1Z.              10254             05/27/80 08:20  05/27/80  [005576]     0
MV2.                7185             01/22/80 23:18  10/31/80  [006233]     0
MV2.RB              1260  D          10/11/80 15:04  10/31/80  [017741]     0
MV3.                6278             01/22/80 23:19  10/31/80  [015576]     0
MV3.RB              1522  D          10/11/80 15:09  10/31/80  [002214]     0
MV8.                5826             01/22/80 23:20  10/31/80  [002200]     0
MV8.RB              1460  D          10/11/80 15:13  10/31/80  [007733]     0
MVAL.               4265             01/22/80 23:23  10/31/80  [006642]     0
MVAL.RB             1010  D          10/29/80 03:33  10/31/80  [017745]     0
MVDT.               3995             01/22/80 23:22  10/31/80  [001164]     0
MVDT.RB             1114  D          10/11/80 15:18  10/31/80  [005540]     0
MVLG.               5008             01/22/80 23:25  10/31/80  [001152]     0
MVLG.RB             1216  D          10/29/80 03:35  10/31/80  [005534]     0
MVMN.               3507             01/22/80 23:25  10/31/80  [020022]     0
MVMN.RB              910  D          10/11/80 15:23  10/31/80  [004433]     0
MVMT.               7691  D          10/23/80 14:57  10/31/80  [006653]     0
MVMT.BU             7696  D          10/21/80 18:35  10/23/80  [003347]     0
MVMT.RB             1840  D          10/29/80 03:44  10/31/80  [017750]     0
MVPV.              15209  D          10/31/80 05:02  10/31/80  [007740]     0
MVPV.BU            15208  D          10/31/80 04:59  10/31/80  [016614]     0
MVPV.RB             4454  D          10/31/80 05:06  10/31/80  [017755]     0
MVPV1.BU           15180  D          10/12/80 01:07  10/12/80  [016663]     0
MVRM.               2179             01/22/80 23:26  10/31/80  [002257]     0
MVRM.RB          628  D        10/29/80 03:37  10/31/80  [001216]    0
MVSP.           9795  D        10/31/80 05:17  10/31/80  [001761]    0
MVSP.BU         9773  D        10/29/80 04:14  10/31/80  [001542]    0
MVSP.RB         2666  D        10/31/80 05:22  10/31/80  [000220]    0
MVTR.           8416           01/22/80 23:27  10/31/80  [000264]    0
MVTR.RB         1692  D        10/29/80 03:40  10/31/80  [001221]    0
MVVB.          15228           05/23/80 20:06  10/31/80  [000370]    0
MVVB.RB        11938  D        10/11/80 16:01  10/31/80  [001274]    0
RLDR.SV         9216  SPWD     12/10/77 19:41  04/25/79  [001640]    0
RTDEB03.RB      7494  PWD/PW   05/09/77 06:29  10/31/80  [001445]    0
RTOS1.LB       12444  PWD/PW   02/24/78 07:06  10/31/80  [000544]    0
RTOS2.LB       18340  PWD/PW   02/24/78 07:07  10/31/80  [001325]    0
RTOSGEN.SV     13312  SD       08/29/79 16:49  09/10/79  [001413]    0
SYS.DR          5632  APWDY    10/25/80 04:30  11/01/80  [000006]    1
SYS.OL                  @:BOOTSYS.OL
SYS.SV                  @:BOOTSYS.SV
R
```

# CHAPTER 3

## PLAYBACK PROGRAM

## 3. PLAYBACK PROGRAM

### 3.1 Introduction

The Playback Program is a real-time FORTRAN IV program designed to perform limited statistical analysis on the recorded data. It takes advantage of multitasking and interrupt techniques to provide an effective tool for data analysis. It is fully automatic during execution, while allowing the operator to gain control at any time.

Features of the Playback Program include:

i) Search Options

The operator may start at the current tape position, in a specified file, or at a specified date and time recorded.

Alternately, the operator may search for alarms which may remain those used in the Acquisition Program, or different alarm levels specified by the operator.

The operator can specify an end date and time at which data processing will halt. If the operator does not specify an end date and time, processing continues till the end of the file. In either case, the data processing operation may be halted at any time to bring the program under operator control.

ii) Printout Options

The operator can choose to monitor between one and ten sensors (inclusive) at any pre-selected rate (up to the sampling rate for a sensor). The recorded values of the sensor are printed out on the terminal printer.

Alternately, the operator can dump the data, in which case all recorded values are printed out.

Irrespective of whether any of the above options is chosen, the program accumulates statistical data which is printed out if required. The statistical printout includes the mean, root-mean-square, and standard deviation.

### 3.2 Program Modules

An overview of the program execution is shown in Figure 6. The program executes in a multitasking environment, and programming has been carried out in modules as listed below.

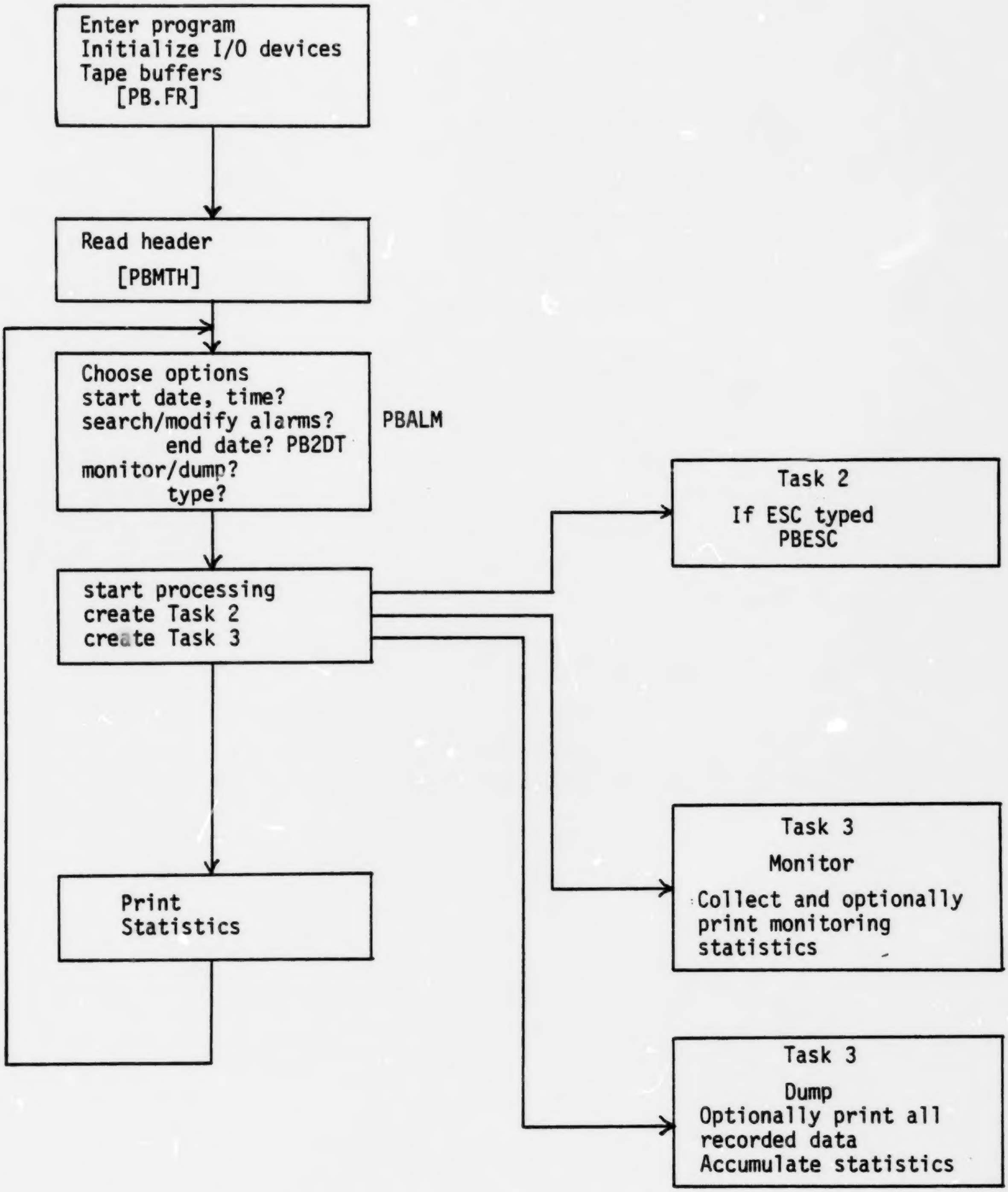
Enter program
Initialize I/O devices
Tape buffers
[PB.FR]
Read header
[PBMTH]
Choose options
start date, time?
search/modify alarms?
end date? PB2DT
monitor/dump?
type?
PBALM
start processing
create Task 2
create Task 3
Print
Statistics
Task 2
If ESC typed
PBESC
Task 3
Monitor
Collect and optionally print monitoring statistics
Task 3
Dump
Optionally print all recorded data
Accumulate statistics

Figure 6 Playback Program - Program Overview

## Modules

| | |
|---|---|
| PB | Initialization, main logic flow |
| PBMN | Task to optionally print monitoring simulation and collect monitoring statistics |
| PBMNF | Subroutine to print monitoring statistics |
| PBDP | Task to optionally print all data |
| PBSTI | Subroutine to initialize statistics |
| PBSTG | Subroutine to gather statistics from tape buffer |
| PBSTT | Subroutine to type accumulated statistics |
| PBMTH | Subroutine to read & process mag tape header |
| PBMTR | Subroutine to read mag tape record |
| PBMTS | Subroutine to check mag tape status |
| PBALM | Subroutine to examine or modify alarm limits |
| PBVAL | Function to convert raw data to engineering units |
| PB2DT | Subroutine to get end date |
| PBESC | Task to abort PBMN or PBDP when ESC typed |
| PBKES | Subroutine to kill wait for ESC |
| IPBYN | Function reads Y or N and returns 1 for Y, 0 for N |
| PBACC | Accept function with error checking |

Labelled Common Areas

| | | |
|---|---|---|
| CHANL | ICTT0 | Channel number for teletype output = 3 |
| | ICTTI | Channel number for teletype output = 2 |
| | ICMT0 | Channel number for tape drive input = 0 or 1 |
| | ICMT1 | Channel number for other tape drive (not used) |
| | JREAD | Command word for mag tape read = 0 |
| | ISTS | Status word from mag tape read |
| | IER | Error code from I/O calls |
| | MWORD | Word count from mag tape read |
| CMAIN | IFL(4) | Mag tape file designator "MTX:XX" |
| | IUN(2) | Mag tape unit designator "MTX" |
| CHEAD | Tape Header Information | |
| | IDATE(6) | Start date & time |
| | ILG | Logging interval |
| | IRM | Recording mode |
| | IHIAD | Highest sensor address (256 or 512) |
| | IDUM | Dummy |
| | ITAB1(0:511) | Rate/Mode/Address information |
| | ITMIN(0:511) | Minimum alarm limits, as recorded or as modified by operator |
| | ITMAX(0:511) | Maximum alarm limits, as recorded or as modified by operator |
| CTREC | Tape Data Record Information | |
| | JDATE(6) | Record date & time |
| | IMS | Flag distinguishes first from second record for a second |
| | ITYP | -1 Recording mode, n>0 for standby |
| | IN(4088) | Data pairs (Adr. Value) |
| | NIN | Number of words in IN |

CSTAT Statistics Accumulation

| | |
|---|---|
| NREC | Number of tape records read |
| CNUM(256) | Count # values read, each sensor |
| CMIN(256) | Count # values less than assigned minimum |
| CMAX(256) | Count # values greater than assigned maximum |
| CMEAN(256) | Sum of values for mean calculation |
| MIN(256) | Smallest value encountered |
| MAX(256) | Largest value encountered |
| RMS(256) | Sum of squares for RMS calculation |

CTPST Pointers for Statistics Printout

| | |
|---|---|
| LADR1 | Lowest address |
| LADR2 | Highest address |
| LINCR | Address loop increment |
| LTFOR | Temporary format indicator |
| LPFOR | Permanent format indicator |
| LALL | Indicates print all statistics |

CFLAG Flags for Various Purposes

| | |
|---|---|
| IFMON | Set while monitoring option is chosen/controls printing of monitoring statistics |
| IFPRO | Reset when a tape data record is read<br>Set when record has been processed (statistics accumulate) |
| IFEOF | Set when EOF encountered on tape data record read |
| IFEOT | Set when EOF encountered on tape when trying to read a header (2 x EOF = EOT) |
| IFTYP | Set if printout is desired; reset to accumulate statistic only |

CDATE Various Date & Time Variables

| | |
|---|---|
| KDATE(6) | Date and time of last record processed (see IFPRO) |
| KDAT1(6) | Desired start date and time |
| KDAT2(6) | Desired end date and time |

CMNF Monitoring Statistics

| | |
|---|---|
| YCNT | Number of values monitored |
| Y(4) | Number of values less than YMIN, monitoring minimum, number of values grater than YMAS, monitoring maximum |
| YMEAN | Sum of values for computing mean |
| YRMS | Sum of squares for computing RMS |
| IMINY | Smallest value encountered |
| IMAXY | Greatest value encountered |
| IADR | Address of sensor being monitored |

CPBMN Constants for Monitoring Printout

| | |
|---|---|
| DRATE(7) | ASCII description of sampling rates 1-7 |
| ASTR(0:1) | ASCII description of modes |

CMTN Sampling Rates

| | |
|---|---|
| CONST(0:7) | Sampling frequencies in Hz |

CVALUE Conversion Constants for PBVAL

| | |
|---|---|
| ITYP(255) | Index to conversion equation for each sensor |
| UNIT(15) | Engineering Units |
| ISUB(32) | Index of units for conversion equations |

### 3.3 Task Allocation

The program has been designed to operate in a multitasking environment. A diagram of the task interaction is shown in Figure 7. The functions of each of the tasks are as follows.

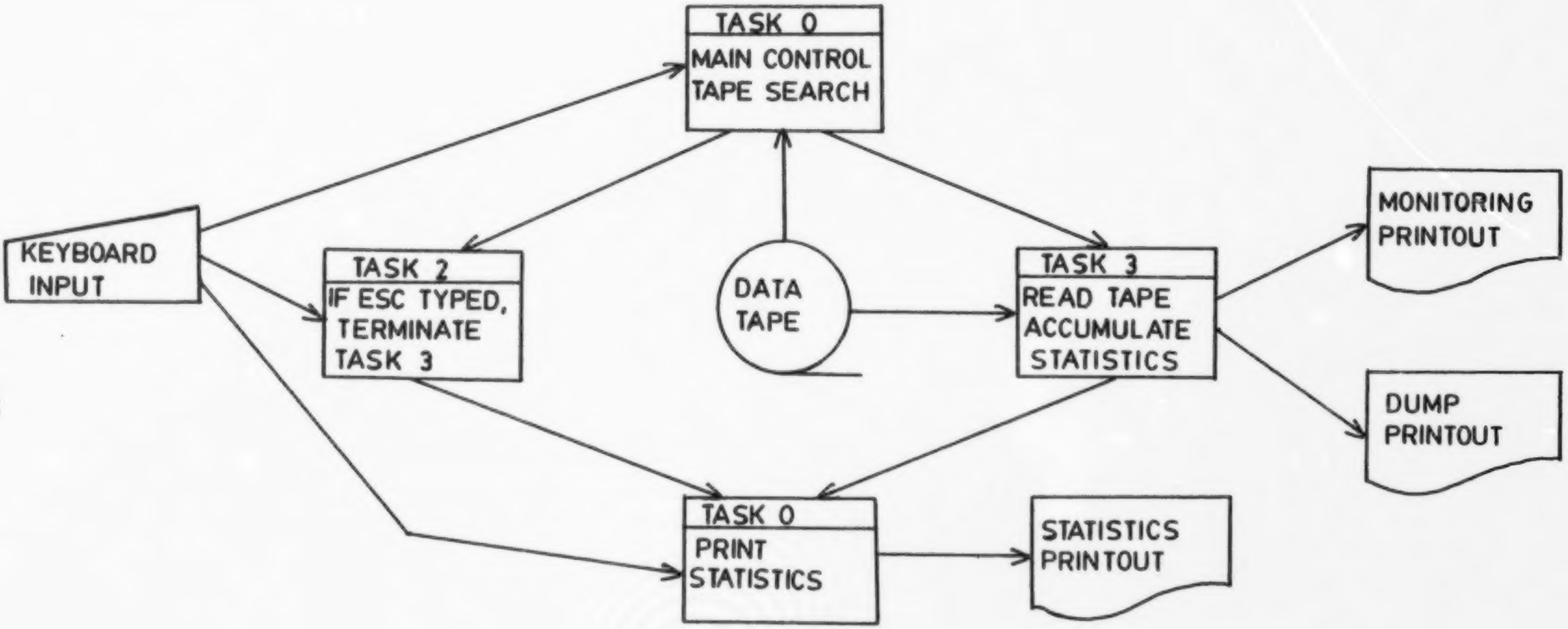
TASK 0
MAIN CONTROL
TAPE SEARCH
KEYBOARD
INPUT
TASK 2
IF ESC TYPED,
TERMINATE
TASK 3
DATA
TAPE
TASK 3
READ TAPE
ACCUMULATE
STATISTICS
MONITORING
PRINTOUT
DUMP
PRINTOUT
TASK 0
PRINT
STATISTICS
STATISTICS
PRINTOUT

FIGURE 7 PLAYBACK PROGRAM — TASK INTEACTION

Task Functions

I. Task 0 - Main Control PROGRAM PB

- A. Initialize
  1. Initialize tape
  2. Initialize file (modify alarms)
  3. Optional start time
  4. Search tape for start condition
  5. Optional end time
  6. Initialize statistics
- B. Process
  1. Select mode (monitor or dump)
  2. Create task 2 to wait for ESC key
  3. Create a task 3 to process/print
  4. Suspend self
- C Print statistics SUBROUTINE PBSTT

II. Task 3 - Printing Options TASK PBMN, TASK PBDP

- A. Initialize - ask if printout desired
- B. Process - read tape, accumulate statistics, optional print
- C. At end
  1. If monitoring, print monitoring statistics
  2. Kill task 2
  3. Ready task 0
  4. Kill self

III. Task 2 - wait for ESC key TASK PBESC

- A. Wait for ESC key
- B. If received, terminate printout
  1. Kill task 3
  2. If monitoring, print monitoring statistics
  3. Ready task 0
  4. Kill self

Tasks are discussed in more detail in Section 3.4 under the module in which they occur.

## 3.4 Program Structure Outline

### Level I Detail

```
1.A.4    Search for start condition                    200
             Read a record
             Check status
             If record
                 If searching for start time           210
                     If time<start time
                         go to 200
                     If time>start time
                         Print "Record Found",
                         date, time
                         go to 300
                     End if
                 If searching for alarm                230
                     Compare each value to alarm
                     If exceeds
                        Print adr, "limit
                        exceeded"
                        go to 300
                     Else
                        go to 200
                     End if
                 End if
         If EOF                                        270
                 Type "EOF"
                 Read header
                 If header
                        go to 180
                 If EOF
                        print "EOT"
                        Type appropriate "not found" message
                        Go to 140
                 End if
         End if
```

## Task Description - PBMN

### Monitoring

U - user
P - program

```
     Set up ESC task
a    U   enters adr to monitor
     P   types rate, mode, alarm limits
         if rate = 0 go to  a
     U   Set alarm limits
         Set scan rate (every nth value)
         Select typing or no
b    P   Reads a record
         checks end time, if selected
         If < = end time
                accumulate statistics for all variables
                look for variable to monitor
                     accumulate statistics on monitored variable
                     type (with flags) if requested
                     if first type this record, type time
                     go to  b
300      If > end time
                type "TIME" current time
                If no typing, type last OK time
                type statistics on monitored variable
                do end routine (option to type all statistics)
                user options
                     Continue monitoring - get new end time, go to  b
                     return to A B or C
400      If EOF
                type "EOF"
                If no typing, type  last OK time
                type statistics on monitored variable
                do end routine (option to type all statistics)
                user options
                     return to A B or C
         If ESC
                type "ESC"
                If no typing, type last OK time
                type statistics on monitored variable
                do end routine (option to type all statistics)
                user options
                     Continue monitoring - go to  b
         End if      return to A B or C
```

## Task Description - PBDP

### Dump

```
    Set up ESC TASK

a   Read a record
    Check end time, if selected
    If < = end time
         accumulate statistics for all variables
         If typing, type buffer (with flags)
         go to  a
    If < end time
         type "TIME" current time
         If no typing, type last OK time
         do end routine
         option to continue, get new end time, go to  a
    If EOF
         type "EOF"
         If no typing,  type last OK time
         do end routine
    If ESC
         type "ESC"
         if no typing, type last OK time
         do end routine
         option to continue, go to  A

    End if
```

## Subroutine to Type Statistics - PBSTT

```
Flags:        LALL      0 ranges entered
                        1 all typed, no repeat
              LPFOR    -1 short default  )          permanent format
                        0 input required )
                        1 long default   )
              LTFOR    -1 short specified)          temporary format
                        0 input required )
                        1 long specified )
              LADRI,LADR2, LINCR
```

Design:

```
           Type # of records
           Initialize:  LALL = 1/LPFOR=0/LTFOR=0/LADR1=1/LADR2=HIADR/LINCR=1
A    110   Ask question
                  If NO return
                  If AS, LPFOR = 1, go type header   D
                  If AL, LPFOR = 1, go type header   D

                  LALL = 0
                  If YS, LPFOR = -1, go ask range   B
                  If YL, LPFOR =  1, go ask range   B
                  If YE, go ask range     B
                  Else, prompt & re-ask   A
120/130/140
  B        Ask for range/return option
140/160/170
  C        If both LTFOR & LPFOR = 0, ask for format
200/S 300/L
  D        Type header
           Type info.
           LTFOR = 0
           If LALL = 0, go to  B
           Else return
```

## Magnetic Tape Header Routine - PBMTH

```
Read header/check status
If EOF
     Type EOT
     Set  IFE0T, IFE0F
     Close tape file
     Release mag tape unit -
     Return IRTN
Else
     IFEOF = 0
     IFEOT = 0
     IFPRO = 1
     Copy to KDATE
     Type date
     Sum samples/sec
     User may check sampling pattern
     User may modify alarm limits
     Return
```

## 3.5 Creating a Save File PB.SV

The following procedure should be followed in order to create a save file of the Playback Program, ready for execution under RTOS.

### i) FORTRAN IV Compilation of Source Files

The FORTRAN IV Compiler translates Fortran language (source program) into machine language (object program). The output is a relocatable binary file. Names of the files to be compiled (using the FORT command) are:

PB.FR
PBDP.FR
PBSTT.FR
PBMTS.FR
PB2DT.FR
IPBYN.FR
PBMN.FR
PBSTI.FR
PBMTH.FR
PBALM.FR
PBESC.FR
PBACC.FR
PBMNF.FR
PBSTG.FR
PBMTR.FR
PBVAL.FR
PBKES.FR

The names are contained in the file 'FILESPB'.

PB<,2DT, ACC, DP, ESC, KES, MTH, MTR, MTS, ALM, MN, MNF, STG, STI, VAL, STT> IPBYNR

### ii) RTOS System Generation (PB.TR)

This procedure will produce a relocatable binary which will trigger the loading of appropriate device drivers from the RTOS library, allocate tables and control blocks used by the system, and allocate a fixed number of channels and task control blocks. Use of the system generation program, RTOSGEN, is shown on the next page. The program is named PB.TR.

```
RTOSGEN

 RTOS  SYSTEM GENERATION REV 1.50

 CPU TYPE(03/02/64) ? 03
  CORE STORAGE (IN K WORDS) 32

  RTC FREQ (0=10HZ,1=100HZ.2=1000HZ.3=LINE) 2

  TASKS(1-255) ? 3
  CHANNELS(1-63) ? 4

RESPOND WITH NUMBER OF UNITS

  DSK(0-1) ? 0
  DKP(0-4) ? 0
  DFL(0-4) ? 0
  DSM(0-4) ? 0
  MTA(0-8) ? 2
    BUFFERED I/O (N/Y) ? N
  PTR(0-2) ? 0
  PTP(0-2) ? 0
  LPT(0-2) ? 0
  CDR(0-2) ? 0
  QTY(0-4) ? 0
  TTYS(0-2) ? 1
  MCA(0-15) ? 0
  AUTO RESTART (Y/N) ? Y


  HIGH PRIORITY INTERRUPTS(N/Y) ? N

  USER SUPPLIED DRIVERS , (N/Y) ? N

     SUMMARY OF RTOS SYSGEN

     CODE      EXTERNAL   NAME

      10       TTIDC      $TTI
      11       TTODC      $TTO
      14       RTCDC
      22       MTADC

   SYSGEN OK (Y/N) ? Y
   OUTPUT FILENAME ? PB.TR
 R
```

iii) Get Libraries

The following libraries are required:

| | |
|---|---|
| FMT.LB | RTOS FORTRAN IV Multitasking |
| FORT1.LB | Standard FORTRAN IV run time libraries |
| FORT2.LB | |
| FORT3.LB | |
| FSYS.LB | Fortran System library |
| SMPYD.LB | Software multiply/divide |
| MTADR.LB | Magnetic Tape support |
| RTOS1.LB | Standard libraries supporting general RTOS functions |
| RTOS2.LB | |

If a debugger is to be included

| | |
|---|---|
| RTDEB03.RB | Standard 1603 RTOS debugger |

iv) Relocatable Load

Using the Relocatable Loader (RLDR command) all relocatable binary files, libraries, and debugger are combined with the RTOSGEN System to create a save file ready to be executed under RTOS. The command string for this operation is contained in file 'LDPB1'

```
RLDR/C/D  @FILESPB@  PB.TR  <FMT, FORT  <1,2,3>,
FSYS, SMPYD, MTADR, RTOS<1,2>>.LB  PB1.SV/S  RTDEB0/3
```

When the debugger is not required use the command string

```
RLDR/C  @FILESPB@  PB.TR  <FMT, FORT  <1,2,3>,
FSYS, SMPYD, MTADR, RTOS<1.2>>.LB   PB1.SV/S
```

The load map thus generated is shown on the next page.

```
@LDPB10
PB1.SV  LOADED BY RLDR REV 05.50 AT 14:00:07 10/31/80
   .MAIN 000440
    PB2DT 026035
    PBACC 026251
    PBDP  026370
    PBESC 027010
    PBKES 027233
    PBMTH 027267
    PBMTR 027755
    PBMTS 030504
    PBALM 031335
    PBMN  032445
    PBMNF 035365
    PBSTG 035604
    PBSTI 037067
    PBVAL 037305
    PBSTT 041403
    IPBYN 044023
    RTOS  044137
    MTI   044367
    ITASK 045144
    FACAL 045171
    FPEND 045213
    ABORT 045220
    FREAD 045227
    THREA 051303
    RDFLD 051347
    READL 051457
    DFTAB 051732
    OPEN  052035
    RESET 052240
    CLOSE 052265
    IOPRE 052340
    ITEST 052761
    BSTRI 053032
    FALOC 053107
    ARYSZ 053127
    FSBR  053145
    CGT   053342
    SMPY  053366
    ATN   053412
    SQRT  053634
    PLY1  053776
    FLIP  054031
    AFRTN 054062
    FL    054067
    STREG 055044
    DFL   055073
    LDREG 056325
    MVBT  056361
    LD0   056420
    ISARE 056437
    FLINK 056452
    RTER  056614
    WRCH  057025
    SDASC 057042
    SASC  057123
    COUT  057157
    LDSTB 057202
```

```
  MOVEF 057232
  MAD   057257
  FARG0 057302
  ARGUM 057335
  FPZER 057405
  FPTRS 057405
  DUMMY 057405
  ARDUM 057405
  RLSE  057406
  INIT  057416
  MTDIO 057427
  SMPYD 057473
  MTADR 057523
  MTSER 057603
  SMTU  060537
  TACAL 060567
  TIDC  060702
  TPRI  061056
  TPEND 061101
  TSVRS 061111
  SMTSK 061204
  TCBMO 061217
  NSAC3 061632
  DUMMY 061632
  INTD  061641
  TTYDR 062074
  RTCDR 062475
  PWRDR 063111
  SYSTE 063312
  BOOT  064344
  RTIN  064467
  SCHED 064743
  GENIO 065066
  IOSER 065641
  ABORT 066404
  RESOL 066537


   NMAX 066537
   ZMAX 000244
   CSZE 000000
    EST 000000
    SST 000000

R
```

The disk directory of all software required to perform various maintenance operations related to the Playback Program is listed on the following pages.

```
DIR DF1
R
LIST/A/E/S
$LPT.               0  RAP       11/01/80 04:35  11/01/80  [0000000]      1
$PTR.               0  APW       11/01/80 04:35  11/01/80  [0000000]      1
$TTI.               0  APW       11/01/80 04:35  11/01/80  [0000000]      1
$TTO.               0  RAP       11/01/80 04:35  11/01/80  [0000000]      1
$TTO.L1             0  D         10/06/80 15:17  10/06/80  [0011163]      0
$TTP.               0  RAP       11/01/80 04:35  11/01/80  [0000000]      1
$TTR.               0  APW       11/01/80 04:35  11/01/80  [0000000]      1
ASM.SV           8944  SPWD      11/30/76 13:57  04/25/79  [0004463]      0
BOOT.SV          6144  SPWD      06/21/78 21:16  04/25/79  [0001333]      0
BOOTSYS.OL      32768  PC        10/05/80 16:16  10/05/80  [0000020]      0
BOOTSYS.SV      46080  SPWD      12/21/77 00:22  09/23/79  [0003346]      0
CLI.ER           4096  PWD       08/25/78 16:58  10/28/80  [0003335]      0
CLI.OL          50176  PWC       08/25/78 16:58  10/28/80  [0001173]      0
CLI.SV          14336  SPWD      08/25/78 16:58  04/25/79  [0001150]      0
COM.CM            369            10/31/80 13:59  10/31/80  [0010065]      0
EDIT.SV          7168  SPWD      11/22/77 11:24  04/25/79  [0000016]      0
FILESPB.           69  D         10/08/80 13:43  10/31/80  [0016627]      0
FILESPB.BU         58  D         10/06/80 00:26  10/08/80  [0011145]      0
FIV.SV          20480  SPWD      12/02/77 10:09  04/25/79  [0012533]      0
FMT.LB           5856  PWD       12/01/77 05:49  10/31/80  [0006672]      0
FORT.SV          2560  SPWD      12/02/77 09:47  04/25/79  [0005506]      0
FORT1.LB        22998  PWD       12/01/77 06:53  10/31/80  [0005514]      0
FORT2.LB        15882  PWD       12/01/77 07:32  10/31/80  [0005572]      0
FORT3.LB        15350  PWD       12/01/77 08:03  10/31/80  [0006333]      0
FSYS.LB          4008  PWD       12/01/77 06:07  10/31/80  [0007707]      0
IPBYN.FR          289            11/02/78 20:13  10/31/80  [0011333]      0
IPBYN.RB          564  D         10/05/80 20:15  10/31/80  [0013403]      0
LDPB1.             80  D         10/06/80 19:29  10/31/80  [0014333]      0
LDPB1.BU           82  D         10/06/80 17:01  10/06/80  [0012226]      0
LFE.SV           8704  SPWD      11/21/77 10:29  04/25/79  [0011233]      0
MAP.DR            152  APWC      10/05/80 16:16  10/10/80  [0000017]      0
MTADR.LB         4164  PWD/PW    02/24/78 15:25  10/31/80  [0010213]      0
PB.BU            4129            10/05/80 18:51  10/05/80  [0013473]      0
PB.FR            4157  D         10/05/80 20:44  10/05/80  [0014713]      0
PB.RB            7422  D         10/05/80 21:09  10/31/80  [0013603]      0
PB.TR            1048            10/31/80 13:58  10/31/80  [0014003]      0
PB1.SV          56320  SD        10/31/80 14:04  10/31/80  [0011043]      0
PB2DT.FR          559            11/02/78 20:11  10/31/80  [0010763]      0
PB2DT.RB         1004  D         10/05/80 19:01  10/31/80  [0010663]      0
PBACC.FR          349            05/04/78 16:31  10/31/80  [0011003]      0
PBACC.RB          580  D         10/05/80 19:44  10/31/80  [0010673]      0
PBALM.FR         1691  D         12/01/78 14:44  10/31/80  [0016073]      0
PBALM.RB         2558  D         10/08/80 13:47  10/31/80  [0011343]      0
PBDP.FR           346            12/02/78 20:58  10/31/80  [0011013]      0
PBDP.RB          1582  D         10/05/80 19:48  10/31/80  [0010703]      0
PBESC.BU          696  D         10/12/80 12:00  10/12/80  [0012473]      0
PBESC.FR          705  D         10/12/80 19:02  10/31/80  [0015563]      0
PBESC.RB         1040  D         10/12/80 19:05  10/31/80  [0010713]      0
PBKES.FR          266            05/04/78 08:21  10/31/80  [0011053]      0
PBKES.RB          302  D         10/05/80 19:53  10/31/80  [0010723]      0
PBMN.BU          4311  D         10/17/80 21:58  10/19/80  [0015723]      0
PBMN.FR          4311  D         10/19/80 20:22  10/31/80  [0016463]      0
PBMN.RB          8106  D         10/19/80 20:29  10/31/80  [0011033]      0
PBMN1.BU         4260  D         10/16/80 19:16  10/16/80  [0016543]      0
PBMN1.RB         8236  D         10/16/80 20:13  10/16/80  [0016573]      0
PBMNF.FR          520            09/26/79 20:38  10/31/80  [0016253]      0
PBMNF.RB         1132  D         10/08/80 13:54  10/31/80  [0016323]      0
```

```
PBMNX.FR         2657         12/01/78 14:33  10/16/80  [0016173]   0
PBMTH.BU         1082  D      10/23/80 15:50  10/23/80  [0012203]   0
PBMTH.FR         1051  D      10/23/80 18:15  10/31/80  [0014023]   0
PBMTH.RB         2348  D      10/23/80 18:19  10/31/80  [0011063]   0
PBMTR.FR         1207         12/02/78 12:36  10/31/80  [0011113]   0
PBMTR.RB         2228  D      10/05/80 19:59  10/31/80  [0012523]   0
PBMTS.FR          874         06/17/78 11:27  10/31/80  [0011143]   0
PBMTS.RB         2334  D      10/05/80 20:02  10/31/80  [0013243]   0
PBSRG.RB            0  D      10/07/80 10:09  10/07/80  [0013613]   0
PBSRG.SR            0  D      10/07/80 10:09  10/07/80  [0013623]   0
PBSTG.BU         1599  D      10/07/80 17:27  10/19/80  [0012343]   0
PBSTG.FR         1599  D      10/19/80 17:37  10/31/80  [0016353]   0
PBSTG.RB         3040  D      10/07/80 17:31  10/31/80  [0011223]   0
PBSTI.FR          474         09/25/79 10:30  10/31/80  [0011203]   0
PBSTI.RB         1034  D      10/05/80 20:05  10/31/80  [0013343]   0
PBSTT.BU         3358  D      10/17/80 19:02  10/17/80  [0012133]   0
PBSTT.FR         3358  D      10/17/80 20:24  10/31/80  [0015653]   0
PBSTT.RB         6028  D      10/17/80 19:09  10/31/80  [0011163]   0
PBVAL.BU         4070  D      10/25/80 00:34  10/31/80  [0012123]   0
PBVAL.FR         4147  D      10/31/80 05:52  10/31/80  [0011103]   0
PBVAL.RB         5918  D      10/31/80 05:56  10/31/80  [0011303]   0
RLDR.SV          9216  SPWD   12/10/77 19:41  04/25/79  [0014273]   0
RTOS1.LB        12444  PWD/PW 02/24/78 07:06  10/31/80  [0007223]   0
RTOS2.LB        18340  PWD/PW 02/24/78 07:07  10/31/80  [0007543]   0
RTOSGEN.SV      13312  SD     08/29/79 16:49  09/10/79  [0010333]   0
SMPYD.LB          200  PWD    12/01/77 08:04  10/31/80  [0007203]   0
SYS.DR           5632  APWDY  10/05/80 15:16  11/01/80  [0000063]   1
SYS.OL                  @:BOOTSYS.OL
SYS.SV                  @:BOOTSYS.SV
R
```

## APPENDICES

A. Listings of Data Acquisition Program Modules

B. Listings of Playback Program Modules

C. Software Back-Up

D. System Diagnostics

E. ROLM Documentation

# APPENDIX A

## DATA ACQUISITION PROGRAM

## LISTINGS

```
0001 MV0   MACRO REV 06.50                  20:59:39 10/31/80
                           .TITL   MV0     ;STORAGE

                           .EXTD   P.CR,R.LN,WRN.A,GT.CH,MT.EF,MT.RL,MT.HD
                           .EXTD   P.NL,P.ER,LGMNB,LGMNC,P.DT
                           .EXTD   IBUF,TAB1A,TAB3A,TAB4A,HIADR,LINCR
                           .EXTD   MTCS,MTNS,MTCCN,TNUM,TNUM0,MNADR
                     ;
                           .EXTN   VBFLG,VB000,VB010,VB020,VB030,VB040,VB060
                           .EXTN   DT000,SP000,AL000,LG000,MN000,RM000
                           .EXTN   TR000,TR100,TR200,MN500
                           .EXTN   .ABORT,.KILL,.PRI,.TASK,.TIDK
                           .EXTN   INDCT,OTDCT,UCLST,MTBFL,MTBHL
                     ;
       000021 IAD1=  21
       000022 IAD3=  22
       000023 IAD4=  23
                     ;
                           .ZREL
                           .ENT    MVINI
000000-000000 MVINI:       .-.     ;0 FOR NO MODS
       000001              .TXTM   1
                           .NREL
                           .ENT    MV001
              MV001:
000000'044101 Q00:         .TXT    /HALT DATA GATHERING (Y OR N)? /
       046124
       020104
       040524
       040440
       043501
       052110
       042522
       044516
       043440
       024131
       020117
       051040
       047051
       037440
       000000
000020'041517 Q01:         .TXT    /CONTINUING HALTS DATA GATHERING, /
       047124
       044516
       052511
       047107
       020110
       040514
       052123
       020104
       040524
       040440
       043501
       052110
       042522
       044516
       043454
       020000
000041'005077 Q02:         .TXT    /<12>? /
       020000
000043'051105 Q03:         .TXT    /READY, /
```

```
 0002 MV0
       040504
       054454
       020000
000047'006400 Q04:    .TXT    /<15>/
000050'005122 Q07:    .TXT    /<12>RECORDING START TIME:/
       042503
       047522
       042111
       047107
       020123
       052101
       051124
       020124
       044515
       042472
       000000
000064'005123 Q05:    .TXT    /<12>SELECT OPTION D,S,A,L,M,R,T,OR HELP<15>/
       042514
       042503
       052040
       047520
       052111
       047516
       020104
       026123
       026101
       026114
       026115
       026122
       026124
       026117
       051040
       044105
       046120
       006400
000107'005105 Q06:    .TXT    /<12>ESC<15>/
       051503
       006400
              ;
              ;       I/O NAMES
              ;
       000056 DCIN=  56
       000057 DCOUT= 57
       000000 MMT0=  0
       000001 MMT1=  1
       000002 MTTI=  2
       000003 MTTO=  3
000112'000226"QTTO:   .*2+2
000113'022124         .TXT    /$TTO/
       052117
       000000
000116'000236"QTTI:   .*2+2
000117'022124         .TXT    /$TTI/
       052111
       000000
              ;
000122'000056 INAC0:  DCIN
000123'177777 INAC1:  INDCT
000124'000057 OTAC0:  DCOUT
000125'177777 OTAC1:  OTDCT
```

```
                      ;
                               .ENT    MTAS
000126'000000 MTAS:   .-.      ;ADDRESS MT BUFFER A
000127'000000 MTBS:   .-.      ;ADDRESS MT BUFFER B
000130'177777 MTFL:   MTBFL    ;LENGTH MT BUFFER
000131'177777 MTHL:   MTBHL    ;LENGTH BUFFER HEADER
                      .TITL    MV0     ;MV000
                      .ENT     MV000
                      ;   INITIALIZE I/O
000132'006017 MV000:  .SYSTM
000133'005000         .RESET
000134'000401          JMP      .+1
                      ;   OPEN TTO
000135'126400          SUB      1,1
000136'020754          LDA      0,QTTO
000137'006017          .SYSTM
000140'014003          .OPEN    MTTO
000141'000400          JMP      .
                      ;   OPEN TTI
000142'020754          LDA      0,QTTI
000143'006017          .SYSTM
000144'014002          .OPEN    MTTI
000145'000400          JMP      .
                      ;   DEFINE INPUT DEVICE
000146'020754          LDA      0,INAC0           ;GET DEVICE CODE
000147'024754          LDA      1,INAC1           ;GET DCT ADR
000150'006017          .SYSTEM
000151'021007          .IDEF
000152'000400          JMP      .
                      ;   DEFINE OUTPUT DEVICE
000153'020751          LDA      0,OTAC0           ;GET DEVICE CODE
000154'024751          LDA      1,OTAC1           ;GET DCT ADR
000155'006017          .SYSTEM
000156'021007          .IDEF
000157'000400          JMP      .
                      ;ONE-TIME INITIALIZATION
000160'020746          LDA      0,MTAS            ;CHECK IF BUFS
000161'101004          MOV      0,0,SZR           ;HAVE BEEN ASSIGNED
000162'000454          JMP      MV020
                      ;   GET BUFFER SPACE
000163'006017          .SYSTEM
000164'001400          .MEM
000165'000401          JMP      .+1               ;ERROR RETURN NEVER
000166'044740          STA      1,MTAS            ;SAVE BUFA START
000167'030742          LDA      2,MTHL
000170'147000          ADD      2,1               ;ADD HEADER LENGTH
000171'044023$         STA      1,MTCS            ;SAVE CURR BUFF START
                      ;
000172'020736          LDA      0,MTFL            ;GET BUFFER LENGTH
000173'006017          .SYSTEM
000174'011000          .MEMI                      ;INCREMENT MEMORY
000175'000400          JMP      .
000176'044731          STA      1,MTBS            ;SAVE BUFB START
000177'147000          ADD      2,1               ;ADD HEADER LENGTH
000200'044024$         STA      1,MTNS            ;SAVE NON-CURR BUFF START
                      ;
000201'006017          .SYSTEM
000202'011000          .MEMI                      ;INCREMENT MEMORY
000203'000400          JMP      .
                      ;   INITIALIZE PATTERN TABLES
```

```
0004 MV0
000204'034016$         LDA     3,TAB1A
000205'054021          STA     3,IAD1
000206'031400          LDA     2,0,3           ;GET DEFAULT RATE/MODE
000207'034017$         LDA     3,TAB3A
000210'054022          STA     3,IAD3
000211'021400          LDA     0,0,3           ;GET DEFAULT MIN
000212'034020$         LDA     3,TAB4A
000213'054023          STA     3,IAD4
000214'025400          LDA     1,0,3           ;GET DEFAULT MAX
000215'034027$         LDA     3,TNUM0
000216'054026$         STA     3,TNUM          ;INITIALIZE SAMPLES/SEC
000217'034021$         LDA     3,HIADR
000220'054022$         STA     3,LINCR
000221'014022$         DSZ     LINCR           ;DECR FOR AUTO INCR
                ;
000222'042022 MV010:   STA     0,@IAD3
000223'046023          STA     1,@IAD4
000224'151400          INC     2,2
000225'052021          STA     2,@IAD1
000226'014022$         DSZ     LINCR
000227'000773          JMP     MV010
                ;******************************
                ;   PATCH TAPE ROUTINES
                       .EXTN   MTADC
000230'034405          LDA     3,AMTAD
000231'020403          LDA     0,K4
000232'041412          STA     0,12,3
000233'000403          JMP     MV020
000234'000004 K4:      4
000235'177777 AMTAD:   MTADC
                ;*****************************
                ;OPTIONAL BRANCH ELSEWHERE
                       .ENT    MV020
000236'002401 MV020:   JMP     @.+1
000237'000240'                 MV030
                ;INITIALIZE DATA GATHERING SYSTEM
                       .ENT    MV030
000240'102520 MV030:   SUBZL   0,0
000241'040000-         STA     0,MVINI         ;ALLOW MODIFICATION
                ;  TASK 3A - DO ROUTINES IN ORDER
000242'024501          LDA     1,TK3A3
000243'004521          JSR     MVTK3
000244'000401          JMP     MV035           ;ESC EXIT
                ;   OPTION TO CHANGE THINGS BEFORE STARTING
000245'006511 MV035:   JSR     @VB3SB
000246'020513          LDA     0,Q03B          ;TYPE "READY"
000247'006003$         JSR     @WRN.A
000250'000402          JMP     .+2             ;NOT READY TO GO
000251'000404          JMP     MV040           ;START DATA GATHERING
                ;  TASK 3B - MODIFY BEFORE START
000252'024470          LDA     1,TK3A2
000253'004511          JSR     MVTK3
000254'000771          JMP     MV035           ;ESC ASKS IF DONE
                ;START NEW DATA GATHERING
                       .ENT    MV040
000255'030025$MV040:   LDA     2,MTCCN
000256'006007$         JSR     @MT.HD          ;WRITE TAPE HEADER
000257'006500          JSR     @AUCLS          ;INITIALIZE USER CLOCK
                ;PRINT RECORDING START TIME
000260'020472          LDA     0,Q07B
```

```
000261'006001$          JSR     @P.CR
000262'020471           LDA     0,QBT
000263'006014$          JSR     @P.DT
                  ;   CREATE TAPE WRITE TASK
000264'020453           LDA     0,TK1WD
000265'024451           LDA     1,TK1AD
000266'177777           .TASK
000267'000400           JMP     .
                  ;   SET PRIORITY TO 2
000270'020464           LDA     0,PRI2
000271'177777           .PRI
                  ;CONTINUE DATA GATHERING
000272'006463 MV050: JSR     @VB1SB            ;TYPE "ESC TO INTERVENE"
                  ;   ACTIVATE MONITORING
000273'020455           LDA     0,TK6WD
000274'024453           LDA     1,TK6AD
000275'000266'          .TASK
000276'000400           JMP     .
                  ;   SET FOR NO MODS
000277'102400           SUB     0,0
000300'040000-          STA     0,MVINI           ;NO MODS BY DEFAULT
                  ;   WAIT FOR ESC CHAR
000301'020457           LDA     0,QRESC
000302'006017           .SYSTEM
000303'021023           .WCHAR
000304'000400           JMP     .
                  ;OPERATOR INTERVENTION REQUESTED
                  ;   KILL OTHER KEYBOARD TASKS
000305'024441           LDA     1,TK4ID           ;ABORT TAPE REQUEST
000306'177777           .ABORT
000307'000401           JMP     .+1
                  ;
000310'024441           LDA     1,TK6ID           ;KILL MONITORING/LOGGING
000311'177777           .TIDK
000312'000401           JMP     .+1
                  ;   TYPE "ESC"
000313'020447           LDA     0,Q06B
000314'006010$          JSR     @P.NL
                  ;   IF MONITORING OR LOGGING B, RESTART LOGGING
000315'126520           SUBZL   1,1
000316'030030$          LDA     2,MNADR
000317'151004           MOV     2,2,SZR
000320'044013$          STA     1,LGMNC
000321'030012$          LDA     2,LGMNB
000322'151004           MOV     2,2,SZR
000323'044013$          STA     1,LGMNC
                  ;   DELETE MONITORING AND LOGGING B
000324'126400           SUB     1,1
000325'044030$          STA     1,MNADR
000326'044012$          STA     1,LGMNB
                  ;  TASK 3C - CREATE OPERATOR TASK
000327'024412 MV060: LDA     1,TK3A1           ;NORMAL INTERVENTION
000330'004434           JSR     MVTK3
000331'000776           JMP     MV060             ;ESC LOOPS BACK
                  ;
                  ;   END OF USER INTERVENTION
                  ;
000332'020000-          LDA     0,MVINI
000333'101004           MOV     0,0,SZR
000334'000721           JMP     MV040
```

```
 0006 MV0
 000335'000735         JMP     MV050
                      ;TAPE WRITE TASK
 000336'177777 TK1AD: TR200
 000337'000401 TK1WD: 1B7+1
 000340'000001 TK1ID: 1
                      ;OPERATOR INTERVENTION TASK
 000341'000412'TK3A1: MV100
 000342'000456'TK3A2: MV130
 000343'000542'TK3A3: MV180
 000344'001403 TK3WD: 3B7+3
 000345'000003 TK3ID: 3
                      ;TAPE CHANGE REQUEST TASK
 000346'000004 TK4ID: 4
                      ;MONITORING TASK
 000347'177777 TK6AD: MN500
 000350'003006 TK6WD: 6B7+6
 000351'000006 TK6ID: 6
 000352'000120"Q07B:   Q07*2
 000353'006400 QBT:    6400    ;CR,0
                      ;
 000354'000002 PRI2:   2
 000355'177777 VB1SB: VB010
 000356'177777 VB3SB: VB030
 000357'177777 AUCLS: UCLST
 000360'000033 QRESC: 33
 000361'000106"Q03B:   Q03*2
 000362'000216"Q06B:   Q06*2
 000363'000116"Q04B:   Q04*2
                       .TITL   MVTK3   ;TASK 3 SUBROUTINE
                      ;        AC1 - TASK START ADDRESS
                      ;        RETURN   - ESC
                      ;        RETURN+1 - DONE OK
 000364'054425 MVTK3: STA      3,MVTA3
 000365'020757         LDA      0,TK3WD
 000366'000275'        .TASK
 000367'000401         JMP      .+1
                      ;    WAIT FOR ESC OR RETURN
 000370'020770         LDA      0,QRESC
 000371'006017         .SYSTEM
 000372'021023         .WCHAR
 000373'000400         JMP      .
                      ;    DETERMINE ESC OR RETURN
 000374'125405         INC      1,1,SNR
 000375'000412         JMP      MV380          ;RETURN TERMINATES OP
                      ;    ESC - KILL KEYBOARD TASKS
 000376'024747         LDA      1,TK3ID
 000377'000306'        .ABORT
 000400'000401         JMP      .+1
                      ;
 000401'024745         LDA      1,TK4ID
 000402'000377'        .ABORT
 000403'000401         JMP      .+1
                      ;    TYPE "ESC"
 000404'020756         LDA      0,Q06B
 000405'006010$        JSR      @P.NL
 000406'000402         JMP      MV390
                      ;    RETURN
 000407'010402 MV380: ISZ       MVTA3
 000410'002401 MV390: JMP       @MVTA3
                      ;
```

```
0007 MV0
000411'000000 MVTA3: .-.
                     .TITL   MV100   ;TASK 3
                     .ENT    MV100
              ;  TASC 3C - OPERATOR INTERVENTION
000412'020000-MV100: LDA     0,MVINI         ;CHECK IF MODS ALLOWED
000413'101004        MOV     0,0,SZR
000414'000442        JMP     MV130           ;IF YES, SKIP QUESTION
              ;  ASK "HALT DATA GATHERING?"
000415'020746        LDA     0,Q04B          ;SKIP A LINE
000416'006001$       JSR     @P.CR
000417'006544        JSR     @VB2SB          ;VERBOSE PROMPT
000420'020556 MV110: LDA     0,Q00B
000421'006001$       JSR     @P.CR
000422'006002$       JSR     @R.LN           ;GET RESPONSE
000423'000527        JMP     MV190           ;JUST CR
000424'020015$       LDA     0,IBUF
000425'006004$       JSR     @GT.CH          ;GET CHARACTER
              ;
000426'020554        LDA     0,QRY
000427'122405        SUB     1,0,SNR
000430'000405        JMP     MV120
              ;
000431'020552        LDA     0,QRN
000432'122405        SUB     1,0,SNR
000433'000423        JMP     MV130
000434'000764        JMP     MV110
              ;   MODIFY
000435'020542 MV120: LDA     0,Q01B
000436'006003$       JSR     @WRN.A
000437'000761        JMP     MV110           ;ABORT
              ;   KILL USER CLOCK
000440'006017        .SYSTEM
000441'021002        .RUCLK
000442'000401        JMP     .+1
              ;   DISABLE INTERFACE
000443'060256        NIOC    DCIN
000444'060257        NIOC    DCOUT
              ;   FINISH TAPE FILE
000445'032025$       LDA     2,@MTCCN
000446'024512        LDA     1,MTEOF
000447'006017        .SYSTEM
000450'023077        .MTDIO  77
              ;   KILL TAPE WRITE TASK
000451'024667        LDA     1,TK1ID
000452'000311'       .TIDK
000453'000401        JMP     .+1
              ;
000454'006011$       JSR     @P.ER           ;PRINT ERROR COUNTS
000455'010000-       ISZ     MVINI           ;OK FOR MODS
                     .ENT    MV130
              ;  TASK 3B - CHANGE BEFORE START
              ;  TSAK 3C - CONTINUED
000456'006506 MV130: JSR     @VB4SB          ;TYPE INTRO
000457'022502        LDA     0,@VBFLA
000460'101004        MOV     0,0,SZR
000461'000403        JMP     MV140
000462'020517        LDA     0,Q05B
000463'006001$       JSR     @P.CR
              ;  TYPE PROMPT
000464'006501 MV140: JSR     @VB6SB
```

```
 0008 MV0
000465'020513          LDA      0,Q02B            ;PROMPT
000466'006001$         JSR      @P.CR
000467'006002$         JSR      @R.LN
000470'000462          JMP      MV190             ;JUST CR - SKIP
000471'020015$         LDA      0,IBUF
000472'006004$         JSR      @GT.CH            ;GET FIRST CHAR
000473'034464          LDA      3,A140            ;GET RETURN ADDRESS
                ;DATE/TIME MODIFICATION
000474'020512          LDA      0,QRD
000475'122405          SUB      1,0,SNR
000476'002470          JMP      @DTSUB
                ;SAMPLING PATTERN
000477'020510          LDA      0,QRS
000500'122405          SUB      1,0,SNR
000501'002466          JMP      @SPSUB
                ;ALARM LIMITS
000502'020506          LDA      0,QRA
000503'122405          SUB      1,0,SNR
000504'002464          JMP      @ALSUB
                ;LOGGING INTERVAL
000505'020504          LDA      0,QRL
000506'122405          SUB      1,0,SNR
000507'002462          JMP      @LGSUB
                ;MONITORING
000510'020502          LDA      0,QRM
000511'122405          SUB      1,0,SNR
000512'002460          JMP      @MNSUB
                ;RECORDING MODE
000513'020500          LDA      0,QRR
000514'122405          SUB      1,0,SNR
000515'002456          JMP      @RMSUB
                ;PROMPTING SELECTION
000516'020467          LDA      0,QRP
000517'122405          SUB      1,0,SNR
000520'000405          JMP      MV145             ;MINIMAL
000521'020463          LDA      0,QRH
000522'122404          SUB      1,0,SZR
000523'000404          JMP      MV150
000524'101400          INC      0,0               ;HELPFUL
000525'042434 MV145:   STA      0,@VBFLA
000526'000736          JMP      MV140
                ;***GET FROM TAPE
                ;***PUT TO TAPE
                ;TAPE ROUTINE
                        .ENT    MV150
000527'020465 MV150:   LDA      0,QRT
000530'122404          SUB      1,0,SZR
000531'000733          JMP      MV140
                ;
000532'020000-         LDA      0,MVINI
000533'101005          MOV      0,0,SNR
000534'000404          JMP      MV155             ;NO MODIFY
                ;    MODIFY - NO TAPE WRITING
000535'030025$         LDA      2,MTCCN
000536'006005$         JSR      @MT.EF
000537'006006$         JSR      @MT.RL
000540'006435 MV155:   JSR      @TR1SB
000541'000723          JMP      MV140
                ;
                ;    TASK 3A - INITIALIZATION ROUTINE
 0009 MV0
                     ;
000542'006420 MV180: JSR      @VB0SB          ;VERBOSE VS MINIMAL PROMPTING
000543'006423        JSR      @DTSUB          ;DATE & TIM INIT
                     ;***READ PATTERN FROM TAPE
000544'006423        JSR      @SPSUB          ;RATE & MODE SELECTION
000545'006423        JSR      @ALSUB          ;ALARM LIMIT SELECTION
000546'006423        JSR      @LGSUB          ;DATA LOGGING
000547'006423        JSR      @MNSUB          ;MONITORING
000550'006423        JSR      @RMSUB          ;RECORDING MODE
                     ;***SAVE PATTERN TO TAPE
000551'006423        JSR      @TR0SB          ;TAPE ROUTINE
                     ;
                     ;    END OF TASK 3 USER INTERVENTION
                     ;
                     .ENT     MV190
000552'102000 MV190: ADC      0,0
000553'006017        .SYSTEM
000554'021023        .WCHAR                   ;TERMINATE ID 0 WAIT
000555'000401        JMP      .+1
000556'177777        .KILL                    ;KILL SELF
                     ;
000557'000464'A140:  MV140
000560'060000 MTEOF: 6B3
000561'177777 VBFLA: VBFLG
000562'177777 VB0SB: VB000
000563'177777 VB2SB: VB020
000564'177777 VB4SB: VB040
000565'177777 VB6SB: VB060
000566'177777 DTSUB: DT000
000567'177777 SPSUB: SP000
000570'177777 ALSUB: AL000
000571'177777 LGSUB: LG000
000572'177777 MNSUB: MN000
000573'177777 RMSUB: RM000
000574'177777 TR0SB: TR000
000575'177777 TR1SB: TR100
000576'000000"Q00B:  Q00*2
000577'000040"Q01B:  Q01*2
000600'000102"Q02B:  Q02*2
000601'000150"Q05B:  Q05*2
                     ;
000602'000131 QRY:   "Y
000603'000116 QRN:   "N
000604'000110 QRH:   "H
000605'000120 QRP:   "P
000606'000104 QRD:   "D
000607'000123 QRS:   "S
000610'000101 QRA:   "A
000611'000114 QRL:   "L
000612'000115 QRM:   "M
000613'000122 QRR:   "R
000614'000124 QRT:   "T
                     ;
                     .TITL    MV0
                     .END     MV000

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
  0010 MV0

A140  000557'       8/07    9/22
AL000 000570' XN    1/09    9/31
ALSUB 000570'       8/19    9/06    9/31
AMTAD 000235'       4/25    4/30
AUCLS 000357'       4/58    6/24
DCIN  000056        2/42    2/57    7/36
DCOUT 000057        2/43    2/59    7/37
DT000 000566' XN    1/09    9/29
DTSUB 000566'       8/11    9/03    9/29
GT.CH 000004$ XD    1/03    7/17    8/06
HIADR 000021$ XD    1/05    4/12
IAD1  000021        1/14    4/02    4/19
IAD3  000022        1/15    4/05    4/16
IAD4  000023        1/16    4/08    4/17
IBUF  000015$ XD    1/05    7/16    8/05
INAC0 000122'       2/57    3/25
INAC1 000123'       2/58    3/26
INDCT 000123' XN    1/12    2/58
K4    000234'       4/26    4/29
LG000 000571' XN    1/09    9/32
LGMNB 000012$ XD    1/04    5/44    5/50
LGMNC 000013$ XD    1/04    5/43    5/46
LGSUB 000571'       8/23    9/07    9/32
LINCR 000022$ XD    1/05    4/13    4/14    4/20
MMT0  000000        2/44
MMT1  000001        2/45
MN000 000572' XN    1/09    9/33
MN500 000347' XN    1/10    6/15
MNADR 000030$ XD    1/06    5/41    5/49
MNSUB 000572'       8/27    9/08    9/33
MTADC 000235' XN    4/24    4/30
MTAS  000126' EN    3/02    3/03    3/37    3/44
MTBFL 000130' XN    1/12    3/05
MTBHL 000131' XN    1/12    3/06
MTBS  000127'       3/04    3/53
MTCCN 000025$ XD    1/06    4/56    7/39    8/54
MTCS  000023$ XD    1/06    3/47
MTEOF 000560'       7/40    9/23
MTFL  000130'       3/05    3/49
MTHL  000131'       3/06    3/45
MTNS  000024$ XD    1/06    3/55
MTTI  000002        2/46    3/22
MTTO  000003        2/47    3/17
MT.EF 000005$ XD    1/03    8/55
MT.HD 000007$ XD    1/03    4/57
MT.RL 000006$ XD    1/03    8/56
MV000 000132' EN    3/08    3/10    9/55
MV001 000000' EN    1/23    1/24
MV010 000222'       4/16    4/21
MV020 000236' EN    3/39    4/28    4/33    4/34
MV030 000240' EN    4/35    4/37    4/38
MV035 000245'       4/43    4/45    4/53
MV040 000255' EN    4/49    4/55    4/56    5/60
MV050 000272'       5/13    6/01
MV060 000327'       5/52    5/54
MV100 000412' EN    6/07    7/02    7/03    7/05
MV110 000420'       7/12    7/26    7/30
MV120 000435'       7/21    7/28
MV130 000456' EN    6/08    7/07    7/25    7/50    7/53
```

```
MV140 000464'      7/56    7/60    8/41    8/48    8/58    9/22
MV145 000525'      8/35    8/40
MV150 000527' EN   8/38    8/45    8/46
MV155 000540'      8/52    8/57
MV180 000542'      6/09    9/02
MV190 000552' EN   7/15    8/04    9/15    9/16
MV380 000407'      6/44    6/58
MV390 000410'      6/56    6/59
MVINI 000000- EN   1/19    1/20    4/39    5/21    5/58    7/05    7/49
                   8/50
MVTA3 000411'      6/33    6/58    6/59    7/01
MVTK3 000364'      4/42    4/52    5/53    6/29    6/33
OTAC0 000124'      2/59    3/31
OTAC1 000125'      2/60    3/32
OTDCT 000125' XN   1/12    2/60
PRI2  000354'      5/10    6/21
P.CR  000001$ XD   1/03    5/01    7/10    7/13    7/58    8/02
P.DT  000014$ XD   1/04    5/03
P.ER  000011$ XD   1/04    7/48
P.NL  000010$ XD   1/04    5/38    6/55
Q00   000000'      1/25    9/37
Q00B  000576'      7/12    9/37
Q01   000020'      1/41    9/38
Q01B  000577'      7/28    9/38
Q02   000041'      1/58    9/39
Q02B  000600'      8/01    9/39
Q03   000043'      1/60    6/26
Q03B  000361'      4/46    6/26
Q04   000047'      2/04    6/28
Q04B  000363'      6/28    7/09
Q05   000064'      2/17    9/40
Q05B  000601'      7/57    9/40
Q06   000107'      2/36    6/27
Q06B  000362'      5/37    6/27    6/54
Q07   000050'      2/05    6/18
Q07B  000352'      4/60    6/18
QBT   000353'      5/02    6/19
QRA   000610'      8/17    9/48
QRD   000606'      8/09    9/46
QRESC 000360'      5/23    6/25    6/38
QRH   000604'      8/36    9/44
QRL   000611'      8/21    9/49
QRM   000612'      8/25    9/50
QRN   000603'      7/23    9/43
QRP   000605'      8/33    9/45
QRR   000613'      8/29    9/51
QRS   000607'      8/13    9/47
QRT   000614'      8/46    9/52
QRY   000602'      7/19    9/42
QTTI  000116'      2/52    3/20
QTTO  000112'      2/48    3/15
RM000 000573' XN   1/09    9/34
RMSUB 000573'      8/31    9/09    9/34
R.LN  000002$ XD   1/03    7/14    8/03
SP000 000567' XN   1/09    9/30
SPSUB 000567'      8/15    9/05    9/30
TAB1A 000016$ XD   1/05    4/01
TAB3A 000017$ XD   1/05    4/04
```

```
0012 MV0

TAB4A 000020$ XD     1/05     4/07
TK1AD 000336'        5/06     6/03
TK1ID 000340'        6/05     7/44
TK1WD 000337'        5/05     6/04
TK3A1 000341'        5/52     6/07
TK3A2 000342'        4/51     6/08
TK3A3 000343'        4/41     6/09
TK3ID 000345'        6/11     6/46
TK3WD 000344'        6/10     6/34
TK4ID 000346'        5/29     6/13     6/50
TK6AD 000347'        5/16     6/15
TK6ID 000351'        5/33     6/17
TK6WD 000350'        5/15     6/16
TNUM  000026$ XD     1/06     4/11
TNUM0 000027$ XD     1/06     4/10
TR000 000574' XN     1/10     9/35
TR0SB 000574'        9/11     9/35
TR100 000575' XN     1/10     9/36
TR1SB 000575'        8/57     9/36
TR200 000336' XN     1/10     6/03
UCLST 000357' XN     1/12     6/24
VB000 000562' XN     1/08     9/25
VB010 000355' XN     1/08     6/22
VB020 000563' XN     1/08     9/26
VB030 000356' XN     1/08     6/23
VB040 000564' XN     1/08     9/27
VB060 000565' XN     1/08     9/28
VB0SB 000562'        9/02     9/25
VB1SB 000355'        5/13     6/22
VB2SB 000563'        7/11     9/26
VB3SB 000356'        4/45     6/23
VB4SB 000564'        7/53     9/27
VB6SB 000565'        7/60     9/28
VBFLA 000561'        7/54     8/40     9/24
VBFLG 000561' XN     1/08     9/24
WRN.A 000003$ XD     1/03     4/47     7/29
.ABOR 000402' XN     1/11     5/30     6/47     6/51
.KILL 000556' XN     1/11     9/20
.PRI  000271' XN     1/11     5/11
.TASK 000366' XN     1/11     5/07     5/17     6/35
.TIDK 000452' XN     1/11     5/34     7/45
```

```
0001 MV1   MACRO REV 06.50            11:58:15 10/31/80
                     .TITL   MV1
       000056 DCIN=  56
       000057 DCOUT= 57
       001000 HIAD=  1000    ;400 FOR 16K SYSTEM
       036000 MKALM= 17B5
       000021 IAD1=  21
       000022 IAD2=  22
       000023 IAD3=  23
                     .ENT    RATEM,RATE0,RATE1,RATE2,RATE3
                     .ENT    RATE4,RATE5,RATE6,RATE7
       042362 RATEM= 17650.
       000000 RATE0= 0
       000001 RATE1= 1       ;1/ 10 SEC
       000012 RATE2= 10.     ;1/ SEC
       000144 RATE3= 100.    ;10/ SEC
       000764 RATE4= 500.    ;50/ SEC
       001750 RATE5= 1000.   ;100/ SEC
       011610 RATE6= 5000.   ;500/ SEC
       023420 RATE7= 10000.  ;1000/ SEC
              ;
                     .ZREL
                     .ENT    TAB1A,TAB3A,TAB4A,TAB1I,ADR,HIADR,INADR
000000-000012'TAB1A: TAB1
000001-001012'TAB3A: TAB3
000002-002012'TAB4A: TAB4
000003-000000 TAB1I: .-.
000004-000000 ADR:   .-.
000005-000000 INADR: .-.
000006-000400 HIADR: 400
                     .ENT    TNUM0,MKADR,TLGA,LGSAV
000007-004766 TNUM0: 377*RATE2       ;# ADRS * DEFAULT RATE
000010-000777 MKADR: 777
000011-003012'TLGA:  TABLG
000012-000000 LGSAV: .-.     ;TABLG ADDR IF SAVING,ELSE ZERO
              ;
                     .ENT    RAT7A,RAT6A,RAT5A,RAT4A,RAT3A,RAT2A,RAT1A
000013-005370'RAT1A: RAT1
000014-004367'RAT2A: RAT2
000015-004106'RAT3A: RAT3
000016-004042'RAT4A: RAT4
000017-004020'RAT5A: RAT5
000020-004014'RAT6A: RAT6
000021-004012'RAT7A: RAT7
              ;
                     .NREL
              ;
              ;      TABLES
              ;
                     .ENT    MTBFL,MTBHL,MTHLN,MTHDR
       010000 MTBFL= 4096.   ;2900. FOR 16K SYSTEM
       000010 MTBHL= 8.
       053012 MTHLN= 3*HIAD+12+5B3   ;TAPE HEADER LENGTH+WRITE COMM
000000'000012 MTHDR: .BLK    12
                     .ENT    TAB1
000012'004000 TAB1:  2B5             ;DEFAULT RATE/MODE
000013'000777        .BLK    HIAD-1  ;RATE,MODE,ADR
                     .ENT    TAB3
001012'000000 TAB3:  0               ;DEFAULT MIN
001013'000777        .BLK    HIAD-1  ;MIN
                     .ENT    TAB4
```

```
 0002 MV1
002012'001777 TAB4:  1777                    ;DEFAULT MAX
002013'000777        .BLK    HIAD-1   ;MAX
                     .ENT    TABLG
003012'001000 TABLG: .BLK    HIAD     ;LOGGING SAVE TABLE
              ;
                     .ENT    RAT7
004012'000001 RAT7:  .BLK    RATEM/RATE7
004013'000000        0
                     .ENT    RAT6
004014'000003 RAT6:  .BLK    RATEM/RATE6
004017'000000        0
                     .ENT    RAT5
004020'000021 RAT5:  .BLK    RATEM/RATE5
004041'000000        0
                     .ENT    RAT4
004042'000043 RAT4:  .BLK    RATEM/RATE4
004105'000000        0
                     .ENT    RAT3
004106'000260 RAT3:  .BLK    RATEM/RATE3      ;HIAD FOR 16K SYSTEM
004366'000000        0
                     .ENT    RAT2
004367'001000 RAT2:  .BLK    HIAD
005367'000000        0
                     .ENT    RAT1
005370'001000 RAT1:  .BLK    HIAD
006370'000000        0
                     .TITL   UCLST
                     .EXTD   LGMNC,LGMNS,LGMNB
                     .EXTN   MTTK1,.UCEX,.IXMT
              ;   CONSTANTS
006371'000012-UAD1A: RAT1A-1
006372'006565'UA1A:  UA1-1
006373'006555'URT1A: URT1-1
006374'006575'ADRIS: UCLIS
006375'177771 KM7:   -7
006376'000000 UAC0:  .-.
006377'000000 UAC1:  .-.
006400'000000 UAC2:  .-.
006401'000000 UAC3:  .-.
              ;
              ;   USER CLOCK START ROUTINE
              ;
                     .ENT    UCLST
006402'040774 UCLST: STA     0,UAC0
006403'044774        STA     1,UAC1
006404'050774        STA     2,UAC2
006405'054774        STA     3,UAC3
              ;   INITIALIZE TAPE BUFFERS
006406'020006$       LDA     0,MTCS
006407'040004$       STA     0,MTC
006410'020007$       LDA     0,MTNS
006411'040005$       STA     0,MTN
              ;   ZERO VARIABLES
006412'102400        SUB     0,0
006413'040542        STA     0,URT0           ;****ZERO CONTINUOUS COUNTER
006414'040022-       STA     0,SP             ;INITIALIZE STACK
006415'040523        STA     0,SPURZ          ;ZERO ERROR COUNT
              ;   START STACK
006416'020523        LDA     0,UAZER
006417'004500        JSR     SPUSH            ;ZERO AMP COMMAND
```

```
 0003 MV1
006420'060277         INTDS
006421'006025-        JSR     @IN.ST            ;START I/O
006422'060177         INTEN
                ;   INITIALIZE COUNTERS
006423'034746         LDA     3,UAD1A           ;ADR OF LIST
006424'054021         STA     3,IAD1
006425'034745         LDA     3,UA1A            ;ADR OF TABLE
006426'054022         STA     3,IAD2
006427'034744         LDA     3,URT1A           ;ADR OF COUNTERS
006430'054023         STA     3,IAD3
                ;
006431'030744         LDA     2,KM7             ;LOOP COUNTER
006432'126520         SUBZL   1,1               ;GET 1 IN AC1
                ;
006433'036021 UST10:  LDA     3,@IAD1           ;GET TABLE ADR
006434'021400         LDA     0,0,3             ;GET FIRST ENTRY
006435'101004         MOV     0,0,SZR           ;CHECK FOR ZERO
006436'161000         MOV     3,0               ;ELSE STORE ADR
006437'042022         STA     0,@IAD2
006440'046023         STA     1,@IAD3           ;INITIALIZE COUNTERS TO 1
006441'151404         INC     2,2,SZR           ;COUNT LOOP
006442'000771         JMP     UST10
006443'046023         STA     1,@IAD3           ;ONE MORE COUNTER
                ;   INITIALIZE LOGGING
006444'125400         INC     1,1               ;GET 2
006445'044001$        STA     1,LGMNC
006446'020011-        LDA     0,TLGA
006447'040012-        STA     0,LGSAV           ;SET TO SAVE LOGGING
                ;   WAIT FOR EVEN 10 SECONDS
                      .ENT    UST20
006450'000401 UST20:  JMP     .+1     ;PATCH UST50 TO SKIP****
006451'006017         .SYSTEM
006452'021003         .GTOD
006453'000012 D10:    10.     ;ERROR RETURN NOT USED
006454'101003         MOV     0,0,SNC
006455'000407         JMP     UST50             ;ZERO SEC
006456'030775         LDA     2,D10
                ;   LOOP TO CHECK
006457'142405 UST30:  SUB     2,0,SNR
006460'000404         JMP     UST50             ;MULT OF 10
006461'101113         MOVL#   0,0,SNC
006462'000775         JMP     UST30
006463'000765         JMP     UST20
                ;   START USER CLOCK
006464'102520 UST50:  SUBZL   0,0               ;GET 1 IN AC0
006465'024707         LDA     1,ADRIS
006466'006017         .SYSTEM
006467'021001         .DUCLK
006470'000401         JMP     .+1
                ;   RESTORE REGISTERS
006471'020705         LDA     0,UAC0
006472'024705         LDA     1,UAC1
006473'030705         LDA     2,UAC2
006474'002705         JMP     @UAC3             ;RETURN
                      .TITL   SPUSH
                      .ENT    SPUSH,S.PUS,SP,STAK,SPURZ
                      .ZREL
000022-000000 SP:     .-.     ;STACK POINTER
000023-006476'STAA:   STAK
000024-006517'S.PUS:  SPUSH
```

```
0004 MV1
                     .NREL
006475'006516'STAE:  STAK+20
006476'000021 STAK:  .BLK    21               ;STACK
              ;
              ;  STACK PUSH ROUTINE - ENTER WITH TABLE ADDRESS IN AC0
              ;
006517'101005 SPUSH: MOV     0,0,SNR          ;CHECK FOR ZERO
006520'001400        JMP     0,3              ;RETURN IMMEDIATELY
006521'054416        STA     3,SPUA3          ;SAVE RETURN ADDRESS
006522'034022-       LDA     3,SP             ;GET STACK POINTER
006523'175004        MOV     3,3,SZR          ;TEST STACK POINTER
006524'175401        INC     3,3,SKP          ;IF NON-ZERO, INCREMENT
006525'034023-       LDA     3,STAA           ;IF ZERO, INITIALIZE
006526'054022-       STA     3,SP             ;SAVE UPDATED STACK POINTER
006527'041400        STA     0,0,3            ;SAVE TABLE ADDRESS IN STACK
006530'020745        LDA     0,STAE           ;CHECK FOR STACK OVERFLOW
006531'116404        SUB     0,3,SZR
006532'002405        JMP     @SPUA3           ;OK - RETURN
              ;ERROR - STACK OVERFLOW
006533'054022-       STA     3,SP             ;ZERO STACK
006534'010404        ISZ     SPURZ            ;COUNT ERRORS
006535'000401        JMP     .+1
006536'002401        JMP     @SPUA3
              ;
006537'000000 SPUA3: .-.
006540'000000 SPURZ: .-.     ;ERROR COUNT, STACK OVERFLOWS
                     .TITL   UCL     ;STORAGE
006541'006542'UAZER: .+1
006542'002000        2000    ;BIT TO ZERO AMPS
006543'000000        0
006544'000000 UMIN:  0       ;MINUTE FLAG
              ;   COUNT CONSTANTS
       023420 UCT=   10000.
006545'023420 UCT1:  UCT/RATE1       ;1/ 10 SEC
006546'001750 UCT2:  UCT/RATE2       ;1/ SEC
006547'000144 UCT3:  UCT/RATE3       ;10/ SEC
006550'000024 UCT4:  UCT/RATE4       ;40/ SEC
006551'000012 UCT5:  UCT/RATE5       ;100/ SEC
006552'000002 UCT6:  UCT/RATE6       ;250/ SEC
006553'000006 UCT60: 6.
006554'000764 UCTWR: 764     ;500 MSEC BETWEEN TAPE WRITES
              ;   COUNTERS
                     .ENT    URT0
006555'000000 URT0:  0
006556'000001 URT1:  1
006557'000001 URT2:  1
006560'000001 URT3:  1
006561'000001 URT4:  1
006562'000001 URT5:  1
006563'000001 URT6:  1
006564'000001 URT60: 1
006565'000001 URTWR: 1
              ;   ADDRESS OR ZERO FOR STACK
006566'000000 UA1:   0
006567'000000 UA2:   0
006570'000000 UA3:   0
006571'000000 UA4:   0
006572'000000 UA5:   0
006573'000000 UA6:   0
006574'000000 UA7:   0
```

```
                        .TITL   UCLIS
                        .ENT    UCLIS
                        .EXTD   MTC,MTN,MTCS,MTNS,MNVAL
                  ;
                  ;     USER CLOCK INTERRUPT SERVICE
                  ;
006575'054541 UCLIS:    STA     3,UCLA3            ;SAVE RETURN ADR
006576'010757           ISZ     URT0               ;****INCREMENT COUNTINUOUS COUNT
006577'000401           JMP     .+1                ;****
                  ;   PROCESSING FOR TAPE WRITE
006600'014765           DSZ     URTWR
006601'000416           JMP     UCL10
006602'020752           LDA     0,UCTWR
006603'040762           STA     0,URTWR
006604'020004$          LDA     0,MTC              ;SAVE END ADR FOR WRITE
006605'040005$          STA     0,MTN
006606'020006$          LDA     0,MTCS             ;SWAP BUFFER POINTERS
006607'024007$          LDA     1,MTNS
006610'044006$          STA     1,MTCS
006611'040007$          STA     0,MTNS
006612'044004$          STA     1,MTC              ;START CURRENT BUFFER
                  ;   ACTIVATE TAPE WRITE TASK
006613'020524           LDA     0,MTMSA
006614'024741           LDA     1,URT0             ;SEND "WHICH HALF SEC"
006615'177777           .IXMT
006616'000401           JMP     .+1
                  ;   START WITH SLOWEST REPEAT
006617'014737 UCL10:    DSZ     URT1               ;DECREMENT COUNTER
006620'000427           JMP     UCL20              ;SKIP UNLESS ZERO
006621'020724           LDA     0,UCT1             ;GET COUNT CONSTANT
006622'040734           STA     0,URT1             ;RESTART COUNTER
006623'020743           LDA     0,UA1              ;GET ZERO OR ADR
006624'004673           JSR     SPUSH              ;PUSH ON STACK
                  ;   CHECK FOR MINUTE
006625'014737           DSZ     URT60
006626'000421           JMP     UCL20
                  ;   PROCESSING FOR ONE MINUTE
006627'020724           LDA     0,UCT60            ;RESTART COUNTER
006630'040734           STA     0,URT60
006631'040713           STA     0,UMIN             ;SET FLAG
                  ;   LOGGING COUNT
006632'020003$          LDA     0,LGMNB            ;CHECK LOGGING B
006633'101004           MOV     0,0,SZR
006634'000404           JMP     UCL15
006635'020002$          LDA     0,LGMNS            ;ZERO FOR NO LOGGING
006636'101005           MOV     0,0,SNR
006637'000410           JMP     UCL20              ;SKIP IF NO LOGGING
                  ;
006640'014001$UCL15:    DSZ     LGMNC              ;DECR MINUTE COUNTER
006641'000406           JMP     UCL20
006642'040001$          STA     0,LGMNC            ;RESET COUNTER
006643'020475           LDA     0,LGMSA            ;MESSAGE ADDR
006644'126520           SUBZL   1,1
006645'006615'          .IXMT
006646'000401           JMP     .+1
                        .ENT    UCL20
                  ;    1-SECOND
006647'014710 UCL20:    DSZ     URT2               ;DECREMENT COUNTER
006650'000407           JMP     UCL30              ;SKIP UNLESS ZERO
006651'020675           LDA     0,UCT2             ;GET COUNT CONSTANT
```

```
0006 MV1
006652'040705        STA    0,URT2       ;RESTART COUNTER
006653'020714        LDA    0,UA2        ;GET ZERO OR ADR
006654'004643        JSR    SPUSH        ;PUSH ON STACK
              ;
006655'102400        SUB    0,0          ;****DEBUG ONLY
006656'040677        STA    0,URT0       ;****RESTART CONTINUOUS COUNTER
              ;
006657'014701 UCL30: DSZ    URT3         ;DECREMENT COUNTER
006660'000415        JMP    UCL40        ;SKIP UNLESS ZERO
006661'020666        LDA    0,UCT3       ;GET COUNT CONSTANT
006662'040676        STA    0,URT3       ;RESTART COUNTER
006663'020705        LDA    0,UA3        ;GET ZERO OR ADR
006664'004633        JSR    SPUSH        ;PUSH ON STACK
              ;   MONITORING TASK
006665'024010$       LDA    1,MNVAL      ;GET VALUE
006666'125005        MOV    1,1,SNR
006667'000406        JMP    UCL40
006670'102400        SUB    0,0
006671'040010$       STA    0,MNVAL      ;ZERO MESSAGE WORD
006672'020447        LDA    0,MNMSA
006673'006645'       .IXMT
006674'000401        JMP    .+1
              ;
006675'014664 UCL40: DSZ    URT4         ;DECREMENT COUNTER
006676'000405        JMP    UCL50        ;SKIP UNLESS ZERO
006677'020651        LDA    0,UCT4       ;GET COUNT CONSTANT
006700'040661        STA    0,URT4       ;RESTART COUNTER
006701'020670        LDA    0,UA4        ;GET ZERO OR ADR
006702'004615        JSR    SPUSH        ;PUSH ON STACK
              ;
006703'014657 UCL50: DSZ    URT5         ;DECREMENT COUNTER
006704'000405        JMP    UCL60        ;SKIP UNLESS ZERO
006705'020644        LDA    0,UCT5       ;GET COUNT CONSTANT
006706'040654        STA    0,URT5       ;RESTART COUNTER
006707'020663        LDA    0,UA5        ;GET ZERO OR ADR
006710'004607        JSR    SPUSH        ;PUSH ON STACK
              ;
006711'014652 UCL60: DSZ    URT6         ;DECREMENT COUNTER
006712'000405        JMP    UCL70        ;SKIP UNLESS ZERO
006713'020637        LDA    0,UCT6       ;GET COUNT CONSTANT
006714'040647        STA    0,URT6       ;RESTART COUNTER
006715'020656        LDA    0,UA6        ;GET ZERO OR ADR
006716'006024-       JSR    @S.PUS       ;PUSH ON STACK
              ;
006717'020655 UCL70: LDA    0,UA7
006720'006024-       JSR    @S.PUS
              ;ZERO AMPS AT ONE MINUTE INTERVALS
006721'020623        LDA    0,UMIN       ;CHECK MINUTE FLAG
006722'101005        MOV    0,0,SNR
006723'000405        JMP    UCL80        ;SKIP MINUTE PROCESSING
006724'102400        SUB    0,0
006725'040617        STA    0,UMIN       ;ZERO FLAG
006726'020613        LDA    0,UAZER      ;ZERO AMP COMMAND
006727'006024-       JSR    @S.PUS
                     .ENT   UCL80
              ;START I/O IF INACTIVE
006730'020005-UCL80: LDA    0,INADR      ;CHECK I/O STATUS
006731'101005        MOV    0,0,SNR
006732'004410        JSR    INST
              ;RETURN TO CALLER
```

```
 0007 MV1
01 006733'126400 UCL90: SUB     1,1                  ;ZERO FOR NO-RESCHEDULING
02 006734'034402        LDA     3,UCLA3
03 006735'177777        .UCEX
04                    ;
05 006736'000000 UCLA3: .-.
06 006737'177777 MTMSA: MTTK1   ;ADDR FOR TAPE WRITE MESSAGE
07 006740'177777 LGMSA: LGMSG   ;ADDR FOR LOGGING MESSAGE
08 006741'000010$MNMSA: MNVAL   ;ADDR FOR MONITORING MESSAGE
10                      .TITL   INST    ;START INTERFACE
11                      .ENT    INST,IN.ST
12                      .EXTN   LGMSG
13                    ;
14                    ;    INTERFACE START ROUTINE
15                    ;       ADDRESS NEXT ADR (IF ANY) ON STACK
16                    ;
17 000025-006742'IN.ST: INST
18                      .NREL
19 006742'054434 INST:  STA     3,INSA3              ;SAVE RETURN ADDRESS
20 006743'034022-       LDA     3,SP                 ;GET STACK POINTER
21 006744'175004        MOV     3,3,SZR
22 006745'000403        JMP     INS10
23                    ;    STACK EMPTY
24 006746'054005-       STA     3,INADR              ;ZERO CURRENT ADR
25 006747'002427        JMP     @INSA3               ;RETURN
26                    ;STACK NOT EMPTY - START INTERFACE
27 006750'040425 INS10: STA     0,INSA0              ;SAVE REGISTER
28 006751'023400        LDA     0,@0,3               ;GET ADR
29 006752'034531        LDA     3,INALM              ;MASK TO CHECK ALARM LINES
30 006753'061157        DOAS    0,DCOUT
31 006754'117405        AND     0,3,SNR              ;NO IN BUSY FOR ALARMS
32 006755'060156        NIOS    DCIN
33 006756'040005-       STA     0,INADR              ;SAVE CURRENT ADR
34                    ;CHECK NEXT STACK ENTRY
35 006757'034022-       LDA     3,SP                 ;GET STACK POINTER AGAIN
36 006760'011400        ISZ     0,3                  ;INCREMENT STACK ENTRY
37 006761'023400        LDA     0,@0,3               ;PREVIEW NEXT ADR
38 006762'101004        MOV     0,0,SZR              ;CHECK FOR ZERO
39 006763'000410        JMP     INS80                ;IF NOT, RETURN
40                    ;ZERO ADR - POP STACK
41 006764'041400        STA     0,0,3                ;**ZERO STACK ENTRY
42 006765'020023-       LDA     0,STAA               ;GET STACK START
43 006766'162405        SUB     3,0,SNR              ;CHECK STACK EMPTY
44 006767'000403        JMP     INS20
45 006770'174400        NEG     3,3                  ;DECREMENT STACK POINTER
46 006771'160000        COM     3,0
47                    ;STACK BECAME EMPTY
48 006772'040022-INS20: STA     0,SP                 ;ZERO POINTER
49                    ;    RETURN
50 006773'020402 INS80: LDA     0,INSA0
51 006774'002402        JMP     @INSA3
52                    ;
53 006775'000000 INSA0: .-.
54 006776'000000 INSA3: .-.
56                      .TITL   CRDIO
57                      .EXTD   MNADR,MNBAD,RMCNT,MTFLG
58                      .EXTN   .UIEX
59                    ;
60                    ;       CARD INPUT ROUTINE
                      ;
```

```
                     .ENT    INDCT
006777'007002'INDCT: INSAV           ;DCTSV - SAVE AREA ADR
007000'177777        -1              ;DCTMS - MASK
007001'007012'       INIS            ;DCTIS - INTERRUPT ADR
007002'000010 INSAV: .BLK    8.
              ;
                     .ENT    INIS
007012'054467 INIS:  STA     3,INIA3         ;SAVE RETURN ADDRESS
007013'060257        NIOC    DCOUT           ;TURN OFF OUT BUSY OR DONE
007014'050464        STA     2,INIA2
007015'034004$       LDA     3,MTC           ;GET BUFFER ADDRESS
              ;
007016'000404        JMP     .+4             ;****
007017'026463        LDA     1,@AURT0        ;****GET CONTINUOUS COUNTER
007020'045400        STA     1,0,3           ;****SAVE IN BUFFER
007021'175400        INC     3,3             ;****
007022'030005-       LDA     2,INADR         ;GET ADR
007023'051400        STA     2,0,3           ;SAVE IN BUFFER
007024'175400        INC     3,3
007025'060656        DIAC    0,DCIN          ;GET DATA
007026'041400        STA     0,0,3           ;SAVE IN BUFFER
007027'175400        INC     3,3
007030'054004$       STA     3,MTC           ;UPDATE BUFFER ADR
              ;GET CARD ADR ONLY IN AC2
007031'024010-       LDA     1,MKADR
007032'133400        AND     1,2
              ;CHECK LIMITS
007033'034001-       LDA     3,TAB3A         ;MIN TABLE
007034'157000        ADD     2,3             ;INDEX
007035'025400        LDA     1,0,3           ;GET MIN
007036'122532        SUBZL#  1,0,SZC
007037'000406        JMP     INI10           ;VAL < MIN
              ;
007040'034002-       LDA     3,TAB4A         ;MAX TABLE
007041'157000        ADD     2,3             ;INDEX
007042'025400        LDA     1,0,3           ;GET MAX
007043'122132        ADCZL#  1,0,SZC
007044'000417        JMP     INI20           ;VAL <= MAX
              ;LIMITS EXCEEDED - SEND ALARM A, TRIGGER RECORDING
007045'024437 INI10: LDA     1,ALMA
007046'065157        DOAS    1,DCOUT
007047'024013$       LDA     1,RMCNT         ;GET RECORDING DURATION
007050'034014$       LDA     3,MTFLG         ;CHECK RECORDING MODE
007051'175123        MOVZL   3,3,SNC         ;SKIP IF RECORDING
007052'044014$       STA     1,MTFLG         ;TRIGGER RECORDING (STANDBY)
007053'060257        NIOC    DCOUT           ;CANCEL INTERRUPT
              ;ALARM B PROCESSING
007054'034012$       LDA     3,MNBAD
007055'156404        SUB     2,3,SZR
007056'000405        JMP     INI20           ;IF NOT RIGHT ADR, SKIP
              ;   SEND ALARM B
007057'024426        LDA     1,ALMB
007060'065157        DOAS    1,DCOUT
007061'101000        MOV     0,0             ;WAIT
007062'060257        NIOC    DCOUT
              ;SEND NEXT CARD ADR
007063'004657 INI20: JSR     INST            ;START NEXT CARD, IF ANY
              ;   LOGGING SAVE
007064'034012-       LDA     3,LGSAV         ;POINTER TO TABLG
007065'175005        MOV     3,3,SNR         ;CHECK FOR ZERO
```

```
000373'020406        LDA     0,Q13B
000374'006007$       JSR     @P.CR
000375'010004-       ISZ     ERCT
000376'020005-       LDA     0,BFPT
000377'002401        JMP     @FMTA3
000400'000000 FMTA3: .-.
000401'000040"Q13B:  Q13.2
                     .TITL   WRNAB    ;WARNING/ABORT
                     .ENT    WRN.A,WRNAB
                     .EXTD   P.CR,P.NL,R.LN,IBUF,GT.CH
              ;WARNING MESSAGE & CHANCE TO ABORT
                     .ZREL
000012-000402'WRN.A: WRNAB
                     .NREL
000402'054414 WRNAB: STA     3,WRNA3
000403'006007$       JSR     @P.CR            ;TYPE WARNING
000404'020413        LDA     0,Q04B
000405'006012$       JSR     @P.NL            ;TYPE "C TO CONTINUE"
000406'006014$       JSR     @R.LN            ;READ A LINE
000407'000406        JMP     WRN10            ;JUST CR - ABORT
000410'020004$       LDA     0,IBUF
000411'006005$       JSR     @GT.CH           ;GET CHARACTER
000412'020406        LDA     0,QRC            ;GET C
000413'122405        SUB     1,0,SNR
000414'010402        ISZ     WRNA3            ;OK - RETURN+1
              ;ABORT - ASK AGAIN
000415'002401 WRN10: JMP     @WRNA3
000416'000000 WRNA3: .-.
000417'001042"Q04B:  Q04*2
000420'000103 QRC:   103
       000001        .TXTM   1
000421'052131 Q04:   .TXT    /TYPE C TO CONTINUE: /
       050105
       020103
       020124
       047440
       041517
       047124
       044516
       052505
       035040
       000000
                     .TITL   PERR     ;PRINT ERRORS
                     .ENT    PERR,P.ER
                     .EXTN   SPURZ,TPOVF
                     .EXTD   BN.AS,P.CR,P.DT
              ;SUBROUTINE TO TYPE ERROR MESSAGES & ZERO COUNTS
                     .ZREL
000013-000434'P.ER:  PERR
                     .NREL
000434'054437 PERR:  STA     3,PERA3
000435'050435        STA     2,PERA2
000436'044433        STA     1,PERA1
000437'040431        STA     0,PERA0
              ;TYPE DATE & TIME
000440'020471        LDA     0,Q07+14         ;GET CR
000441'006015$       JSR     @P.DT
              ;TYPE STACK RE-ZERO MESSAGE
000442'026433        LDA     1,@ASPRZ
000443'125005        MOV     1,1,SNR
```

```
000444'000407        JMP     PER10
000445'020431        LDA     0,Q06B
000446'006013$       JSR     @BN.AS
000447'020427        LDA     0,Q06B
000450'006007$       JSR     @P.CR
000451'126400        SUB     1,1
000452'046423        STA     1,@ASPRZ
              ;TYPE TAPE BUFFER OVERFLOW
000453'026421 PER10: LDA     1,@ATPOV
000454'125005        MOV     1,1,SNR
000455'000407        JMP     PER20
000456'020421        LDA     0,Q07B
000457'006013$       JSR     @BN.AS
000460'020417        LDA     0,Q07B
000461'006007$       JSR     @P.CR
000462'126400        SUB     1,1
000463'046411        STA     1,@ATPOV
              ;
              PER20:
000464'020404 PER90: LDA     0,PERA0
000465'024404        LDA     1,PERA1
000466'030404        LDA     2,PERA2
000467'002404        JMP     @PERA3
              ;
000470'000000 PERA0: .-.
000471'000000 PERA1: .-.
000472'000000 PERA2: .-.
000473'000000 PERA3: .-.
000474'177777 ATPOV: TPOVF
000475'177777 ASPRZ: SPURZ
000476'001200"Q06B:  Q06*2
000477'001232"Q07B:  Q07*2
       000001        .TXTM   1
000500'054130 Q06:   .TXT    /XXXXXX STACK RE-ZEROES.<15>/
       054130
       054130
       020123
       052101
       041513
       020122
       042455
       055105
       051117
       042523
       027015
       000000
000515'054130 Q07:   .TXT    /XXXXXX BUFFER OVERFLOWS.<15>/
       054130
       054130
       020102
       052506
       043105
       051040
       047526
       042522
       043114
       047527
       051456
       006400
```

```
 0009 MV3
01                          .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
  0010 MV3

ADR    000003$ XD     1/03      3/36      4/30      5/11      5/37
ASPRZ  000475'        7/59      8/07      8/30
ATPOV  000474'        8/09      8/17      8/29
BFPT   000005- EN     1/07      1/13      2/58      3/02      3/14      3/19      4/10
                      4/20      4/24      6/60      7/04
BN.AS  000013$ XD     4/31      5/13      5/56      6/05      7/46      8/03      8/13
CKC80  000362'        6/39      6/44
CKC90  000364'        6/42      6/46
CKCA1  000370'        6/35      6/45      6/51
CKCA2  000367'        6/34      6/46      6/50
CKCA3  000366'        6/33      6/44      6/47      6/49
CKCR   000351' EN     6/22      6/23      6/31      6/33
CK.CR  000010- EN     6/23      6/31
D1023  000253'        5/03      5/48
DIG.   000006$ XD     1/04      3/09      4/06
ERCT   000004- EN     1/07      1/12      2/37      3/01      3/17      3/26      4/23
                      7/03
FMTA3  000400'        6/59      7/05      7/06
FMTER  000371' EN     6/41      6/52      6/53      6/57      6/59
FM.ER  000011- EN     3/50      6/53      6/57
GAD10  000124'        2/56      3/04
GAD30  000141'        3/12      3/20
GAD40  000142'        2/49      3/07      3/22
GAD50  000153'        3/33      3/39
GAD60  000161'        3/24      3/41
GAD70  000167'        3/43      3/48
GAD90  000173'        3/28      3/34      3/46      3/53
GADA3  000174'        2/35      3/52      3/53      3/55
GADR   000075' EN     2/24      2/25      2/33      2/35      3/60
GTA10  000210'        4/08      4/14
GTA20  000215'        4/15      4/20
GTA30  000217'        4/12      4/22
GTA90  000222'        4/18      4/26
GTAA3  000223'        4/05      4/26      4/27
GTAD   000201' EN     2/40      2/51      4/01      4/05
GT.CH  000005$ XD     1/04      2/46      3/04      3/20      6/24      6/36      7/10
                      7/22
G.ADR  000006- EN     2/25      2/33
HIADR  000001$ XD     1/03      4/16
IBUF   000004$ XD     1/03      2/38      7/10      7/21
IFMOD  000003- EN     1/07      1/11      3/41
LADR1  000000- EN     1/07      1/08      2/41      2/54      3/31
LADR2  000001- EN     1/07      1/09      2/42      2/52      3/32
LINCR  000002- EN     1/07      1/10      2/44      3/10      3/30
MAX    000254'        5/04      5/44      6/03
MKMOD  000251'        5/01      5/27
MKRAT  000252'        5/02      5/19
MV003  000000' EN     1/17      1/18
PER10  000453'        8/01      8/09
PER20  000464'        8/11      8/19
PER90  000464'        8/20
PERA0  000470'        7/54      8/20      8/25
PERA1  000471'        7/53      8/21      8/26
PERA2  000472'        7/52      8/22      8/27
PERA3  000473'        7/51      8/23      8/28
PERR   000434' EN     7/43      7/44      7/49      7/51
P.CR   000007$ XD     1/04      2/60      3/16      3/45      4/22      4/31      5/34
                      6/54      7/02      7/10      7/16      7/46      8/05      8/15
P.DT   000015$ XD     7/46      7/57
```

```
  0011 MV3

P.ER   000013- EN     7/44     7/49
P.NL   000012$ XD     4/31     5/53     5/59     6/01     6/08     6/11     7/10
                      7/18
Q00    000224'        4/38     4/51     4/52     5/25     5/31
Q00B   000240'        4/51     5/12
Q00B3  000241'        4/52     5/33
Q01    000232'        4/44     4/53
Q01B   000242'        4/53     5/52     5/60
Q02    000233'        4/45     4/54     4/55
Q02B   000243'        4/54     5/55     6/04
Q02B2  000244'        4/55     5/58     6/07
Q03    000237'        4/49     4/56
Q03B   000245'        4/56     6/10
Q04    000421'        7/29     7/32
Q04B   000417'        7/17     7/29
Q06    000500'        8/31     8/34
Q06B   000476'        8/02     8/04     8/31
Q07    000515'        7/56     8/32     8/47
Q07B   000477'        8/12     8/14     8/32
Q11    000000'        1/19     2/19
Q11B   000070'        2/19     4/11
Q12    000007'        1/26     2/17     2/20
Q12B   000071'        2/20     4/21
Q12.2  000016" EN     2/16     2/17
Q13    000020'        1/35     2/18
Q13B   000401'        7/01     7/07
Q13.2  000040" EN     2/16     2/18     7/07
Q14    000027'        1/42     2/21
Q14B   000072'        2/21     2/59
Q15    000036'        1/49     2/22
Q15B   000073'        2/22     3/15
Q17    000050'        1/59     2/23
Q17B   000074'        2/23     3/44
QB0CM  000250'        4/60     5/23
QLP    000247'        4/59     5/28
QLR    000246'        4/58     5/30
QRC    000420'        7/23     7/30
QRCM   000200'        3/48     3/59
QRCR   000177'        3/22     3/58     6/37
QRDS   000175'        2/47     3/56
QRSL   000176'        3/05     3/57
RPS    000255' EN     4/28     4/29     4/35     5/05     5/06
RPS10  000322'        5/47     5/52
RPS30  000337'        5/50     6/10
RPSA0  000345'        5/09     6/13     6/18
RPSA1  000346'        5/08     6/14     6/19
RPSA2  000347'        5/07     6/15     6/20
RPSA3  000350'        5/06     6/16     6/21
R.LN   000014$ XD     7/10     7/19
R.PS   000007- EN     3/37     4/29     4/35
SPURZ  000475' XN     7/45     8/30
TAB1A  000002$ XD     1/03     4/30     5/15
TAB3A  000010$ XD     4/30     5/38
TAB4A  000011$ XD     4/30     5/41
TPOVF  000474' XN     7/45     8/29
WRN10  000415'        7/20     7/27
WRNA3  000416'        7/15     7/25     7/27     7/28
WRNAB  000402' EN     7/08     7/09     7/13     7/15
```

```
 0012 MV3
WRN.A 000012- EN    7/09    7/13
```

```
0001 MV8   MACRO REV 06.50                  13:23:00 10/31/80
                        .TITL     MV8      ;TEST
                        .ENT      TEST
                  ;
                  ;     TEST MODE 0 - LOOP
                  ;
                        .ZREL
000000-060477 TEST:     READS     0
000001-000002-          JMP       .+1
000002-040036-          STA       0.TEHI
000003-062677 TEST0:    IORST
000004-175440           SUBO      3,3
000005-152440           SUBO      2,2
000006-024034-          LDA       1,TEALM
000007-061157           DOAS      0,DCOUT
000010-107405           AND       0,1,SNR
000011-060156           NIOS      DCIN
                  ;WAIT FOR IN CLK
000012-063456 TEST1:    SKPBN     DCIN
000013-000016-          JMP       TEST2
000014-151400           INC       2,2
000015-000012-          JMP       TEST1
                  ;WAIT FOR EXT GATE
000016-063457 TEST2:    SKPBN     DCOUT
000017-000022-          JMP       TEST3
000020-175400           INC       3,3
000021-000016-          JMP       TEST2
                  ;READ INPUT
000022-060257 TEST3:    NIOC      DCOUT
000023-064656           DIAC      1,DCIN
                  ;DELAY
000024-034035-          LDA       3,TECNT
000025-175404           INC       3,3,SZR
000026-000025-          JMP       .-1
                  ;INCR ADR
000027-101400           INC       0,0
000030-024036-          LDA       1,TEHI
000031-106432           SUBZ#     0,1,SZC
000032-000003-          JMP       TEST0
000033-000000-          JMP       TEST
                  ;
000034-036000 TEALM:    17B5
000035-176000 TECNT:    176000
000036-000377 TEHI:     377
                        .ENT      PATCH
000037-000040 PATCH:    .BLK      40
                        .TITL     MV8      ;MV800
                        .ENT      MV800,MV801,MV820,MV840,MV850,MV860
                        .ENT      ITAP,OPUT,OP020
                        .ENT      RTC03,RTC02,MTCNT,TYCNT,TAB,PTCH1
                        .ENT      BUFS,BUFE
                  ;
                        .EXTN     .UCEX,URT0,MV000,MTAS
                        .EXTD     P.CR,R.LN,BN.AS,MTC,INADR,S.PUS,IN.ST
                  ;
       000000 MMT0=     0
       000002 MTTI=     2
       000056 DCIN=     56
       000057 DCOUT=    57
                        .ZREL
000077-000000 BUFS:     .-.
```

```
000100-000000 BUFE:   .-.
                      .NREL
       000001         .TXTM   1
                MV801:
000000'052101 M2:     .TXT    /TAPE? /
       050105
       037440
       000000
000004'052131 M4:     .TXT    /TYPE? /
       050105
       037440
       000000
000010'000002 MTU:    .BLK    2
000012'000004 MTF:    .BLK    4
000016'000020"MTUB:   MTU*2   ;TAPE UNIT BYTE POINTER
000017'000024"MTFB:   MTF*2   ;TAPE FILE BYTE POINTER
000020'177400 MKL:    377B7
000021'000000"MSG2:   M2*2
000022'000010"MSG4:   M4*2
000023'177777 AMTAS:  MTAS
              ;
000024'022777 MV800:  LDA     0,@AMTAS          ;GET BUFFER AREA
000025'040100-        STA     0,BUFE
000026'040077-        STA     0,BUFS
000027'040004$        STA     0,MTC
              ;ASK IF TAPE OUTPUT
000030'020771         LDA     0,MSG2            ;BYTE POINTER
000031'006001$        JSR     @P.CR             ;WRITE MESSAGE
              ;   WAIT FOR RESPONSE - NAME IF DESIRED
000032'020765         LDA     0,MTFB            ;CR IF NOT
000033'006017         .SYSTM
000034'015402         .RDL    MTTI
000035'000400         JMP     .
000036'125225         MOVZR   1,1,SNR           ;TEST FOR JUST CR
000037'000434         JMP     MV830
              ;   BREAK OUT TAPE NAME
000040'020752         LDA     0,MTF             ;TRANSFER FIRST WORD
000041'040747         STA     0,MTU
000042'020751         LDA     0,MTF+1           ;ALTER SECOND WORD
000043'024755         LDA     1,MKL
000044'123400         AND     1,0               ;MASK OUT SECOND CHAR
000045'040744         STA     0,MTU+1
              ;
000046'020747         LDA     0,MTF+3           ;CHECK 1- OR 2-DIGIT FILE
000047'101004         MOV     0,0,SZR
000050'000405         JMP     MV810
              ;        1-DIGIT
000051'020743         LDA     0,MTF+2
000052'123400         AND     1,0
000053'040741         STA     0,MTF+2
000054'000403         JMP     MV820
              ;        2-DIGIT
000055'102400 MV810:  SUB     0,0
000056'040737         STA     0,MTF+3
              ;    OPEN OUTPUT FILE
000057'126400 MV820:  SUB     1,1
000060'020736         LDA     0,MTUB
000061'006017         .SYSTEM
000062'004000         .INIT
000063'000400         JMP     .
```

```
000064'020733         LDA     0,MTFB
000065'006017         .SYSTM
000066'025000         .MTOPD  MMT0
000067'000735         JMP     MV800
                ;
000070'020465         LDA     0,K402
000071'042462         STA     0,@AP010
000072'000403         JMP     MV840
000073'020463 MV830:  LDA     0,K401                ;NO TAPE OUTPUT
000074'042457         STA     0,@AP010
                ;
                ;    DECIDE WHETHER OR NOT TO TYPE
                ;    ENTER CR IF NO;  ANY KEY + CR IF YES
                ;
000075'020461 MV840:  LDA     0,K401                ;DEFAULT NO TYPE
000076'042456         STA     0,@AP020
                ;    DECIDE WHETHER TO TYPE
000077'020723         LDA     0,MSG4                ;BYTE POINTER TO MESSAGE
000100'006001$        JSR     @P.CR                 ;TYPE MESSAGE
000101'006002$        JSR     @R.LN                 ;GET RESPONSE
000102'000403         JMP     MV850                 ;CR ONLY
000103'020452         LDA     0,K402                ;USE TTY
000104'042450         STA     0,@AP020
                ;INITIALIZE CLOCK
000105'102400 MV850:  SUB     0,0
000106'042411         STA     0,@RTC03
000107'020412         LDA     0,RTC01
000110'040410         STA     0,RTC02
000111'102520         SUBZL   0,0                   ;GENERATE CONS 1
000112'024411         LDA     1,ADRMV
000113'006017         .SYSTEM
000114'021001         .DUCLK
000115'000401         JMP     .+1
000116'000423         JMP     MV870
                ;
000117'177777 RTC03:  URT0
000120'000000 RTC02:  .-.
000121'000010 RTC01:  10
000122'000000 RTFL:   .-.
                ;
                ;    USER CLOCK INTERRUPT ROUTINE
                ;
000123'000124'ADRMV:  UCLMV
                ;
000124'012773 UCLMV:  ISZ     @RTC03
000125'000401         JMP     .+1
000126'014772         DSZ     RTC02
000127'000404         JMP     UCL10
                ;
000130'020771         LDA     0,RTC01
000131'040767         STA     0,RTC02
000132'040770         STA     0,RTFL
                ;
000133'126520 UCL10:  SUBZL   1,1                   ;GENERATE CONS 1
000134'177777         .UCEX
                ;
                ;         TEST MODE 2 - ADDRESS N GAUGES AT SAME INTERVAL
                ;
000135'020765 MV860:  LDA     0,RTFL                ;CHECK IF TIME TO ADDRESS
000136'101005         MOV     0,0,SNR
```

```
000137'000776          JMP     MV860              ;LOOP & WAIT FOR INTERRUPT
                  ;
000140'004522          JSR     OPUT               ;OUTPUT TO TAPE OR TYPE
000141'102400 MV870:   SUB     0,0
000142'040760          STA     0,RTFL             ;RESET FLAG
000143'020414          LDA     0,TABA             ;GET TABLE ADDRESS
000144'060277          INTDS
000145'004006$         JSR     S.PUS              ;PUSH ONTO STACK
000146'020005$         LDA     0,INADR
000147'101005          MOV     0,0,SNR
000150'004007$         JSR     IN.ST
000151'060177          INTEN
000152'000763          JMP     MV860
                  ;
000153'000273'AP010:   OP010
000154'000315'AP020:   OP020
000155'000402 K402:    402       ;JMP .+2
000156'000401 K401:    401       ;JMP .+1
                  ;
000157'000160'TABA:    TAB
000160'001760 TAB:     1760
000161'001761          1761
000162'001762          1762
000163'001763          1763
000164'001764          1764
000165'001765          1765
000166'001766          1766
000167'001767          1767
000170'001770          1770
000171'001771          1771
000172'001772          1772
000173'001773          1773
000174'001774          1774
000175'001775          1775
000176'001776          1776
000177'001777          1777
000200'000000          0
000201'000040 PTCH1:   .BLK    40
                       .TITL   MV8     ;INTAPE
000241'020715 ITAP:    LDA     0,K401  ;NO TAPE WRITE
000242'040431          STA     0,OP010
                  ;
000243'020077-         LDA     0,BUFS             ;ADDRESS
000244'126400          SUB     1,1                ;READ
000245'006017          .SYSTM
000246'023000          .MTDIO  MMT0
000247'000401          JMP     .+1
                  ;
000250'024410          LDA     1,ITRD             ;CHECK FOR TAPE READ STATUS
000251'132415          SUB#    1,2,SNR
000252'000415          JMP     OPUT+5             ;GO TYPE
                  ;
000253'024406          LDA     1,ITEOF            ;CHECK FOR EOF
000254'132414          SUB#    1,2,SZR
000255'000400          JMP     .                      ;ERROR
000256'002401          JMP     @.+1               ;RESTART
000257'177777                  MV000
                  ;
000260'104103 ITRD:    104103
000261'104503 ITEOF:   104503
```

```
                        .TITL    MV8      ;OUTPUT
000262'014466 OPUT:     DSZ      TYCNT              ;COUNT CALLS
000263'001400           JMP      0,3                ;IF NOT, RETURN
                  ;
000264'054461           STA      3,OP091            ;SAVE RETURN ADDRESS
000265'020462           LDA      0,TYNUM
000266'040462           STA      0,TYCNT
000267'020004$          LDA      0,MTC
000270'040100-          STA      0,BUFE
000271'020077-          LDA      0,BUFS             ;RESTORE POINTER
000272'040004$          STA      0,MTC
000273'000402 OP010:    JMP      .+2                ;OPTIONAL WRITE TO TAPE
000274'000421           JMP      OP020
000275'020077-          LDA      0,BUFS             ;WORD ADDRESS
000276'024450           LDA      1,BUFN             ;WORD COUNT
000277'006017           .SYSTM
000300'023000           .MTDIO   MMT0               ;WRITE TO TAPE
000301'000400           JMP      .                  ;ERROR RETURN
000302'014450           DSZ      MTCNT              ;COUNT TAPE WRITES
000303'000412           JMP      OP020
                  ;
000304'020445           LDA      0,MTNUM
000305'040445           STA      0,MTCNT            ;RESTART COUNTER
                  ;
000306'024445           LDA      1,MTEOF            ;WRITE EOF
000307'006017           .SYSTEM
000310'023000           .MTDIO   MMT0
000311'000401           JMP      .+1
                  ;
000312'024442           LDA      1,CKEOF            ;CHECK STATUS
000313'132414           SUB#     1,2,SZR
000314'000400           JMP      .                          ;ERROR
                  ;
000315'000401 OP020:    JMP      .+1                ;OPTIONAL CONVERSION FOR PRINT O
000316'000425           JMP      OP090
000317'030004$          LDA      2,MTC              ;GET BUFFER ADDR
                  ;
000320'000405           JMP      .+5
000321'020434           LDA      0,PRTB
000322'025000           LDA      1,0,2              ;GET COUNTER
000323'006003$          JSR      @BN.AS
000324'151400           INC      2,2
                  ;
000325'020431           LDA      0,PRTB7
000326'025000           LDA      1,0,2              ;GET ADR
000327'006003$          JSR      @BN.AS
000330'151400           INC      2,2
                  ;
000331'020426           LDA      0,PRTBV
000332'025000           LDA      1,0,2              ;GET VALUE
000333'006003$          JSR      @BN.AS
000334'050004$          STA      2,MTC
                  ;
000335'020421           LDA      0,PRTB7
000336'006001$          JSR      @P.CR
                  ;
000337'024100-          LDA      1,BUFE             ;LOOP FOR ANOTHER LINE
000340'132433           SUBZ#    1,2,SNC
000341'000754           JMP      OP020
                  ;
```

```
000342'020077-         LDA     0,BUFS          ;RESTORE POINTER
000343'040004$         STA     0,MTC
000344'002401 OP090:   JMP     @OP091          ;RETURN
000345'000000 OP091:   .-.
000346'050000 BUFN:    5B3
000347'000003 TYNUM:   3
000350'000003 TYCNT:   3
000351'000010 MTNUM:   10
000352'000010 MTCNT:   10
000353'060000 MTEOF:   6B3
000354'104501 CKEOF:   104501
000355'000740"PRTB:    PRT*2
000356'000747"PRTB7:   PRT*2+7
000357'000756"PRTBV:   PRT*2+16
000360'054130 PRT:     .TXT    /XXXXXX XXXXXX -XX.XX<15>/
       054130
       054130
       020130
       054130
       054130
       054040
       026530
       054056
       054130
       006400
                       .TITL   MV8     ;TT000
                       .ENT    TT110,TT120,TT130,TT140,TT150
                       .ENT    TT160,TT001,TT002
                ;
       000001          .TXTM   1
                       .NREL
              TT001:
000373'046524 MT0:     .TXT    /MT0/
       030000
000375'046524 MT1:     .TXT    /MT1/
       030400
000377'046524 MT2:     .TXT    /MT2/
       031000
000401'046524 MT0.0:   .TXT    /MT0:0/
       030072
       030000
000404'046524 MT1.0:   .TXT    /MT1:0/
       030472
       030000
000407'046524 MT2.0:   .TXT    /MT2:0/
       031072
       030000
                ;
              TT002:
000412'000000 PT0:     0
000413'000766"         MT0*2
000414'001002"         MT0.0*2
000415'000001 PT1:     1
000416'000772"         MT1*2
000417'001010"         MT1.0*2
000420'000002 PT2:     2
000421'000776"         MT2*2
000422'001016"         MT2.0*2
000423'000412'T0:      PT0
000424'000415'         PT1
```

```
0007 MV8
000425'000420'         PT2
000426'000000          0
000427'000423'AT0:     T0
000430'000430'WADR:    .
000431'050000 TWRI:    5B3
000432'060000 TEOF:    6B3
000433'010000 TREW:    1B3
000434'000000 TCHAN:   .-.
                ;INIT
000435'034772 TT110:   LDA      3,AT0
000436'031400 TT111:   LDA      2,0,3
000437'151005          MOV      2,2,SNR
000440'000412          JMP      TT120
000441'021001          LDA      0,1,2
000442'126400          SUB      1,1
000443'124000          COM      1,1
000444'054016          STA      3,USP
000445'006017          .SYSTEM
000446'004000          .INIT
000447'000401          JMP      .+1
000450'175400          INC      3,3
000451'000765          JMP      TT111
                ;OPEN
000452'034755 TT120:   LDA      3,AT0
000453'031400 TT121:   LDA      2,0,3
000454'151005          MOV      2,2,SNR
000455'000412          JMP      TT130
000456'021002          LDA      0,2,2
000457'126400          SUB      1,1
000460'031000          LDA      2,0,2
000461'054016          STA      3,USP
000462'006017          .SYSTEM
000463'025077          .MTOPD   77
000464'000401          JMP      .+1
000465'175400          INC      3,3
000466'000765          JMP      TT121
                ;WRITE DATA
000467'020741 TT130:   LDA      0,WADR
000470'024741          LDA      1,TWRI
000471'030721          LDA      2,PT0
000472'006017          .SYSTEM
000473'023077          .MTDIO   77
000474'000401          JMP      .+1
000475'000772          JMP      TT130
                ;WRITE 2 EOF'S
000476'034731 TT140:   LDA      3,AT0
000477'031400 TT141:   LDA      2,0,3
000500'151005          MOV      2,2,SNR
000501'000416          JMP      TT150
000502'031000          LDA      2,0,2
000503'050731          STA      2,TCHAN
000504'054016          STA      3,USP
000505'024725          LDA      1,TEOF
000506'006017          .SYSTEM
000507'023077          .MTDIO   77
000510'054016          STA      3,USP
000511'030723          LDA      2,TCHAN
000512'024720          LDA      1,TEOF
000513'006017          .SYSTEM
000514'023077          .MTDIO   77
```

```
 0008 MV8
01 000515'175400          INC     3,3
02 000516'000761          JMP     TT141
03                 ;CLOSE
04 000517'034710 TT150:  LDA     3,AT0
05 000520'031400 TT151:  LDA     2,0,3
06 000521'151005          MOV     2,2,SNR
07 000522'000410          JMP     TT160
08 000523'031000          LDA     2,0,2
09 000524'054016          STA     3,USP
10 000525'006017          .SYSTEM
11 000526'014477          .CLOSE  77
12 000527'000401          JMP     .+1
13 000530'175400          INC     3,3
14 000531'000767          JMP     TT151
15                 ;RELEASE
16 000532'034675 TT160:  LDA     3,AT0
17 000533'031400 TT161:  LDA     2,0,3
18 000534'151005          MOV     2,2,SNR
19 000535'000700          JMP     TT110
20 000536'021001          LDA     0,1,2
21 000537'054016          STA     3,USP
22 000540'006017          .SYSTEM
23 000541'002400          .RLSE
24 000542'000401          JMP     .+1
25 000543'175400          INC     3,3
26 000544'000767          JMP     TT161
                          .TITL   MV8
29                        .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
0009 MV8

 DRMV 000123'          3/30     3/43
 MTAS 000023'          2/20     2/22
AP010 000153'          3/07     3/10     4/15
AP020 000154'          3/16     3/23     4/16
AT0   000427'          7/03     7/10     7/24     7/46     8/04     8/16
BN.AS 000003$ XD       1/53     5/41     5/46     5/51
BUFE  000100- EN       1/50     2/01     2/23     5/09     5/57
BUFN  000346'          5/15     6/05
BUFS  000077- EN       1/50     1/60     2/24     4/43     5/10     5/14     6/01
CKEOF 000354'          5/30     6/11
DCIN  000056           1/16     1/18     1/29     1/57
DCOUT 000057           1/14     1/23     1/28     1/58
INADR 000005$ XD       1/53     4/09
IN.ST 000007$ XD       1/53     4/11
ITAP  000241' EN       1/48     4/40
ITEOF 000261'          4/53     4/60
ITRD  000260'          4/49     4/59
K401  000156'          3/09     3/15     4/18     4/40
K402  000155'          3/06     3/22     4/17
M2    000000'          2/05     2/18
M4    000004'          2/09     2/19
MKL   000020'          2/17     2/40
MMT0  000000           1/55     3/03     4/46     5/17     5/27
MSG2  000021'          2/18     2/27
MSG4  000022'          2/19     3/18
MT0   000373'          6/33     6/51
MT0.0 000401'          6/39     6/52
MT1   000375'          6/35     6/54
MT1.0 000404'          6/42     6/55
MT2   000377'          6/37     6/57
MT2.0 000407'          6/45     6/58
MTAS  000023' XN       1/52     2/20
MTC   000004$ XD       1/53     2/25     5/08     5/11     5/36     5/52     6/02
MTCNT 000352' EN       1/49     5/19     5/23     6/09
MTEOF 000353'          5/25     6/10
MTF   000012'          2/14     2/16     2/37     2/39     2/44     2/48     2/50
                       2/54

MTFB  000017'          2/16     2/30     3/01
MTNUM 000351'          5/22     6/08
MTTI  000002           1/56     2/32
MTU   000010'          2/13     2/15     2/38     2/42
MTUB  000016'          2/15     2/57
MV000 000257' XN       1/52     4/57
MV800 000024' EN       1/47     2/22     3/04
MV801 000000' EN       1/47     2/04
MV810 000055'          2/46     2/53
MV820 000057' EN       1/47     2/51     2/56
MV830 000073'          2/35     3/09
MV840 000075' EN       1/47     3/08     3/15
MV850 000105' EN       1/47     3/21     3/25
MV860 000135' EN       1/47     3/59     4/01     4/13
MV870 000141'          3/34     4/04
OP010 000273'          4/15     4/41     5/12
OP020 000315' EN       1/48     4/16     5/13     5/20     5/34     5/59
OP090 000344'          5/35     6/03
OP091 000345'          5/05     6/03     6/04
OPUT  000262' EN       1/48     4/03     4/51     5/02
PATCH 000037- EN       1/44     1/45
PRT   000360'          6/12     6/13     6/14     6/15
```

```
 0010 MV8

PRTB   000355'        5/39    6/12
PRTB7  000356'        5/44    5/54    6/13
PRTBV  000357'        5/49    6/14
PT0    000412'        6/50    6/59    7/40
PT1    000415'        6/53    6/60
PT2    000420'        6/56    7/01
PTCH1  000201' EN     1/49    4/38
P.CR   000001$ XD     1/53    2/28    3/19    5/55
RTC01  000121'        3/27    3/38    3/50
RTC02  000120' EN     1/49    3/28    3/37    3/47    3/51
RTC03  000117' EN     1/49    3/26    3/36    3/45
RTFL   000122'        3/39    3/52    3/59    4/05
R.LN   000002$ XD     1/53    3/20
S..PUS 000006$ XD     1/53    4/08
T0     000423'        6/59    7/03
TAB    000160' EN     1/49    4/20    4/21
TABA   000157'        4/06    4/20
TCHAN  000434'        7/08    7/51    7/57
TEALM  000034-        1/13    1/41
TECNT  000035-        1/31    1/42
TEHI   000036-        1/09    1/36    1/43
TEOF   000432'        7/06    7/53    7/58
TEST   000000- EN     1/02    1/07    1/39
TEST0  000003-        1/10    1/38
TEST1  000012-        1/18    1/21
TEST2  000016-        1/19    1/23    1/26
TEST3  000022-        1/24    1/28
TREW   000433'        7/07
TT001  000373' EN     6/28    6/32
TT002  000412' EN     6/28    6/49
TT110  000435' EN     6/27    7/10    8/19
TT111  000436'        7/11    7/22
TT120  000452' EN     6/27    7/13    7/24
TT121  000453'        7/25    7/36
TT130  000467' EN     6/27    7/27    7/38    7/44
TT140  000476' EN     6/27    7/46
TT141  000477'        7/47    8/02
TT150  000517' EN     6/27    7/49    8/04
TT151  000520'        8/05    8/14
TT160  000532' EN     6/28    8/07    8/16
TT161  000533'        8/17    8/26
TWRI   000431'        7/05    7/39
TYCNT  000350' EN     1/49    5/02    5/07    6/07
TYNUM  000347'        5/06    6/06
UCL10  000133'        3/48    3/54
UCLMV  000124'        3/43    3/45
URT0   000117' XN     1/52    3/36
WADR   000430'        7/04    7/38
.UCEX  000134' XN     1/52    3/55
```

```
0001 PVAL  MACRO REV 06.50         17:45:59 10/31/80
                     .TITL  PVAL    ;PRINT CONVERTED VALUES
                     .ENT   PVAL,P.VAL
                     .EXTD  TAB3A,TAB4A,P.NL,MUL.U,DIV.
                     .ZREL
000000-000000'P.VAL: PVAL
                     .ENT    IFVLT,IFLOG
000001-000000 IFVLT: 0
000002-000000 SVAC2: .-.
000003-000000 SVAC3: .-.
000004-000000 POS:   .-.
000005-000000 IFLOG: .-.
              ;
              ;PRINT IN VOLTS OR APPROPRIATE UNITS
              ;     AC1 - VALUE (0-1023)
              ;     AC2 - ADR
                     .NREL
       000001        .TXTM   1
000000'054464 PVAL:  STA     3,PVAA3
000001'050462        STA     2,PVAA2          ;ADR
000002'044460        STA     1,PVAA1          ;VALUE
000003'040456        STA     0,PVAA0
             ;CHECK ADC ERROR
000004'020463        LDA     0,BIT5
000005'123405        AND     1,0,SNR
000006'000404        JMP     PVA10
000007'106400        SUB     0,1              ;REMOVE ERROR BIT
000010'020462        LDA     0,QSTST          ;**
000011'000416        JMP     PVA40
             ;CHECK LIMITS
000012'034001$PVA10: LDA     3,TAB3A
000013'157000        ADD     2,3
000014'021400        LDA     0,0,3
000015'106532        SUBZL#  0,1,SZC
000016'000406        JMP     PVA20
             ;
000017'034002$       LDA     3,TAB4A
000020'157000        ADD     2,3
000021'021400        LDA     0,0,3
000022'106132        ADCZL#  0,1,SZC
000023'000403        JMP     PVA30
             ;
000024'020447 PVA20: LDA     0,QSTBL          ;*,SP
000025'000402        JMP     PVA40
             ;NO FLAGS
000026'020446 PVA30: LDA     0,QBLBL          ;SP,SP
000027'040451 PVA40: STA     0,QANS+3
             ;CHECK FOR "ALL IN VOLTS"
000030'020001-       LDA     0,IFVLT
000031'101005        MOV     0,0,SNR
000032'000405        JMP     PVA50
000033'030433        LDA     2,AVOLT
000034'050002-       STA     2,SVAC2
000035'007002        JSR     @2,2
000036'000415        JMP     PVA80
             ;OTHER CONVERSIONS
                     .ENT    PVA50
000037'030424 PVA50: LDA     2,PVAA2          ;GET SENSOR ADR
000040'034425        LDA     3,ATAD           ;GET TABLE ADDR
000041'157000        ADD     2,3              ;INDEX
000042'031400        LDA     2,0,3            ;GET POINTER
```

```
0002 PVAL
000043'050002-          STA     2,SVAC2
000044'007002           JSR     @2,2             ;CONVERT
                ;
000045'020005-          LDA     0,IFLOG
000046'101005           MOV     0,0,SNR
000047'000404           JMP     PVA80
                ;    LOGGING
000050'030002-          LDA     2,SVAC2
000051'021001           LDA     0,1,2            ;GET UNITS BYTE POINTER
000052'000402           JMP     .+2
                ;    CR
000053'020415 PVA80:    LDA     0,QCRB
000054'006003$          JSR     @P.NL
                ;RETURN
000055'020404 PVA90:    LDA     0,PVAA0
000056'024404           LDA     1,PVAA1
000057'030404           LDA     2,PVAA2
000060'002404           JMP     @PVAA3
                ;STORAGE
000061'000000 PVAA0:    .-.
000062'000000 PVAA1:    .-.
000063'000000 PVAA2:    .-.
000064'000000 PVAA3:    .-.
000065'001364'ATAD:     STAB
000066'000767'AVOLT:    SVOLT
000067'002000 BITS:     1B5
000070'000162"QCRB:     QCR*2
000071'006400 QCR:      15B7
000072'025052 QSTST:    "**400+"*
000073'025040 QSTBL:    "**400+"
000074'020040 QBLBL:    " *400+"
000075'020040 QANS:     .TXT     /     0.** /
       020040
       030056
       025052
       020000
                        .TITL   CANS     ;CONVERT ANSWER TO ASCII
                        .ENT    CANS,C.ANS
                        .ZREL
000006-000102'C.ANS:    CANS
                        .NREL
000102'020511 CANS:     LDA     0,QBPT1          ;INITIALIZE BYTE POINTER
000103'040506           STA     0,QBPT
000104'020511           LDA     0,APTEN          ;INITIALIZE POWERS OF 10
000105'040507           STA     0,ADTEN
000106'020766           LDA     0,QBLBL          ;INITIALIZE ANSWER
000107'040766           STA     0,QANS
000110'040766           STA     0,QANS+1
000111'020506           LDA     0,QB0DC
000112'040765           STA     0,QANS+2
000113'102400           SUB     0,0              ;INITIALIZE LEFT ZERO SUPPRESS
000114'040473           STA     0,IFRST
000115'030002-          LDA     2,SVAC2          ;GET # DIGITS BEFORE DECIMAL
000116'021003           LDA     0,3,2
000117'101123           MOVZL   0,0,SNC
000120'000406           JMP     CAN10
000121'030004-          LDA     2,POS            ;FLIP SIGN OF ANSWER
000122'151005           MOV     2,2,SNR
000123'152001           ADC     2,2,SKP
000124'152400           SUB     2,2
```

```
000125'050004-         STA     2,POS
000126'101220 CAN10:   MOVZR   0,0
000127'040461          STA     0,NDIG
000130'101004          MOV     0,0,SZR          ;IF ZERO DIGITS
000131'000404          JMP     CAN30
000132'004477          JSR     FIRST            ;DO SIGN IF NECESSARY
000133'020465          LDA     0,QDEC
000134'004516          JSR     STBT             ;STORE DECIMAL
000135'034457 CAN30:   LDA     3,ADTEN
000136'021400          LDA     0,0,3            ;GET POWER OF 10
000137'101005          MOV     0,0,SNR
000140'000426          JMP     CAN80            ;IF ZERO, DONE
000141'152401          SUB     2,2,SKP          ;ACCUMULATE COUNT IN 2
000142'151400 CAN40:   INC     2,2
000143'106422          SUBZ    0,1,SZC
000144'000776          JMP     CAN40
000145'107000          ADD     0,1              ;ONE TOO MANY SUBTRACTED
000146'020441          LDA     0,IFRST          ;IF NO FIRST CHAR
000147'143004          ADD     2,0,SZR          ;AND ASCII 0
000150'000403          JMP     CAN50
000151'010440          ISZ     QBPT
000152'000405          JMP     CAN60
              ;DIGIT FOUND
000153'004456 CAN50:   JSR     FIRST
000154'020446          LDA     0,QR0
000155'143000          ADD     2,0
000156'004474          JSR     STBT
000157'014431 CAN60:   DSZ     NDIG
000160'000404          JMP     CAN70
              ;INSERT DECIMAL
000161'004450          JSR     FIRST
000162'020436          LDA     0,QDEC
000163'004467          JSR     STBT
              ;LOOP INCR
000164'010430 CAN70:   ISZ     ADTEN
000165'000750          JMP     CAN30
              ;END OF CONVERSION
              ;   CHECK -. CASE
000166'020711 CAN80:   LDA     0,QANS+2
000167'024427          LDA     1,QMIDC
000170'122404          SUB     1,0,SZR
000171'000403          JMP     CAN85
000172'020425          LDA     0,QB0DC
000173'040704          STA     0,QANS+2
000174'020412 CAN85:   LDA     0,IFTHR          ;CHECK FOR OPTIONAL
000175'101005          MOV     0,0,SNR          ;   BLANK LAST DIGIT
000176'000404          JMP     CAN90
000177'020424          LDA     0,QRBL           ;YES, BLANK IT
000200'014411          DSZ     QBPT
000201'004451          JSR     STBT
000202'020410 CAN90:   LDA     0,QBPT0
000203'006003$         JSR     @P.NL
              ;RETURN
000204'034003-         LDA     3,SVAC3
000205'001400          JMP     0,3
              ;STORAGE
                       .ENT    IFTHR
000206'000000 IFTHR:   0       ;SET NON-ZERO TO SUPPRESS LAST CHAR
000207'000000 IFRST:   .-.
000210'000000 NDIG:    .-.
```

```
0004 PVAL
000211'000000 QBPT:   .-.
000212'000172"QBPT0:  QANS*2
000213'000173"QBPT1:  QANS*2+1
000214'000000 ADTEN:  .-.
000215'000224'APTEN:  PTEN
000216'026456 QMIDC:  "-*400+".
000217'030056 QB0DC:  "0*400+".
000220'000056 QDEC:   ".
000221'000055 QMIN:   "-
000222'000060 QR0:    "0
000223'000040 QRBL:   "
000224'001750 PTEN:   1000.
000225'000144         100.
000226'000012         10.
000227'000001         1
000230'000000         0
                      .TITL    FIRST    ;FIRST DIGIT
                      .ENT     FIRST
000231'054417 FIRST:  STA      3,FIRA3
000232'034755         LDA      3,IFRST
000233'175004         MOV      3,3,SZR
000234'002414         JMP      @FIRA3
                ;
000235'034004-        LDA      3,POS
000236'175004         MOV      3,3,SZR
000237'000406         JMP      FIR10
                ;NEGATIVE - INSERT SIGN
000240'040411         STA      0,FIRA0
000241'014750         DSZ      QBPT
000242'020757         LDA      0,QMIN
000243'004407         JSR      STBT
000244'020405         LDA      0,FIRA0
                ;RESET LEFT ZERO SUPPRESS
000245'176520 FIR10:  SUBZL    3,3
000246'054741         STA      3,IFRST
000247'002401         JMP      @FIRA3
                ;STORAGE
000250'000000 FIRA3:  .-.
000251'000000 FIRA0:  .-.
                      .TITL    STBT     ;STORE A BYTE
                      .ENT     STBT
                ;STORE A BYTE
                ;  QBPT - BYTE POINTER
                ;  AC0 - BYTE IN RIGHT HALF
000252'054426 STBT:   STA      3,STBA3
000253'050424         STA      2,STBA2
000254'044422         STA      1,STBA1
000255'040420         STA      0,STBA0
000256'030423         LDA      2,QRMK             ;MASK OUT LEFT BYTE
000257'143400         AND      2,0
                ;GET WORD ADDRR
000260'034731         LDA      3,QBPT
000261'175223         MOVZR    3,3,SNC
000262'101301         MOVS     0,0,SKP            ;LEFT BYTE
000263'030417         LDA      2,QLMK             ;RIGHT BYTE
                ;UPDATE
000264'025400         LDA      1,0,3              ;GET EXISTING WORD
000265'147400         AND      2,1                ;MASK OUT BYTE TO BE REPLACED
000266'107000         ADD      0,1                ;ADD IN NEW BYTE
000267'045400         STA      1,0,3              ;STORE
```

```
                   ;RETURN
000270'010721            ISZ     QBPT
000271'020404            LDA     0,STBA0
000272'024404            LDA     1,STBA1
000273'030404            LDA     2,STBA2
000274'002404            JMP     @STBA3
                   ;STORAGE
000275'000000 STBA0:  .-.
000276'000000 STBA1:  .-.
000277'000000 STBA2:  .-.
000300'000000 STBA3:  .-.
000301'000377 QRMK:   377
000302'177400 QLMK:   377B7
                         .TITL   CONVERT
                   ;SUBROUTINES TO CONVERT 0 TO 1023 TO LINEAR FUNCTIONS
                   ;  AC1 - X, 0 TO 1023
                   ;  AC2 - POINTER TO CONVERSION BLOCK, ALSO IN SVAC2
                   ;
                   ; Y=AX/1024
                         .ENT    LINP
000303'054003-LINP:      STA     3,SVAC3
000304'004505            JSR     LIN1               ;AX/1024
000305'002006-           JMP     @C.ANS
                   ;Y=AX/1024 + C
                         .ENT    LINC
000306'054003-LINC:      STA     3,SVAC3
000307'004502            JSR     LIN1               ;AX/1024
000310'004522            JSR     LIN2               ;+C, SET SIGN, CHECK FLIP
000311'002006-           JMP     @C.ANS
                   ;Y=AX/1024 + C; ALTERNATE UNITS
                         .ENT    LINU
000312'054003-LINU:      STA     3,SVAC3
000313'004476            JSR     LIN1
000314'004516            JSR     LIN2
000315'004535            JSR     LIN9
000316'002006-           JMP     @C.ANS
                   ;Y=AX/1024 * C : WINDOWS
000317'054003-LINW:      STA     3,SVAC3
000320'004471            JSR     LIN1               ;AX/1024
000321'004524            JSR     LIN3               ;*C
000322'002006-           JMP     @C.ANS
                   ;FOUR OPEN/CLOSED VALVE SETTINGS
                         .ENT    BUBBL
000323'054003-BUBBL:     STA     3,SVAC3
                   ;    ROUND OFF
000324'020441            LDA     0,C20
000325'107000            ADD     0,1
000326'020444            LDA     0,C1740
000327'107400            AND     0,1
                   ;    CONVERT
000330'020662            LDA     0,QBPT0
000331'040660            STA     0,QBPT
000332'020671            LDA     0,QRBL
000333'004717            JSR     STBT
                   ;
000334'020666            LDA     0,QR0              ;DIGIT 0
000335'030434            LDA     2,C400
000336'133404            AND     1,2,SZR
000337'101400            INC     0,0
000340'004712            JSR     STBT               ;ONE OR ZERO
```

```
                      ;
000341'020661          LDA     0,QR0             ;DIGIT 0
000342'030426          LDA     2,C200
000343'133404          AND     1,2,SZR
000344'101400          INC     0,0
000345'004705          JSR     STBT              ;ONE OR ZERO
                      ;
000346'020654          LDA     0,QR0             ;DIGIT 0
000347'030420          LDA     2,C100
000350'133404          AND     1,2,SZR
000351'101400          INC     0,0
000352'004700          JSR     STBT              ;ONE OR ZERO
                      ;
000353'020647          LDA     0,QR0             ;DIGIT 0
000354'030412          LDA     2,C40
000355'133404          AND     1,2,SZR
000356'101400          INC     0,0
000357'004673          JSR     STBT              ;ONE OR ZERO
                      ;
000360'020643          LDA     0,QRBL
000361'004671          JSR     STBT
000362'020630          LDA     0,QBPT0
000363'006003$         JSR     @P.NL
000364'002003-         JMP     @SVAC3
                      ;CONSTANTS
000365'000020 C20:    20
000366'000040 C40:    40
000367'000100 C100:   100
000370'000200 C200:   200
000371'000400 C400:   400
000372'001740 C1740:  1740
                        .ENT    POLY2,POLY3
                        .ENT    POLSG,POL2,POL3,POLX,POLY   ;DEBUG****
                      ;SECOND ORDER POLYNOMIAL
000373'054003-POLY2:   STA     3,SVAC3
000374'004505          JSR     POL2              ;0-2ND ORDER TERMS
000375'004472          JSR     POLSG             ;SIGN
000376'030002-         LDA     2,SVAC2           ;CHECK IF UNITS CHANGE W/ SIG
000377'021006          LDA     0,6,2
000400'101004          MOV     0,0,SZR
000401'004451          JSR     LIN9              ;ALTERNATE UNITS
000402'002006-         JMP     @C.ANS
                      ;THIRD ORDER POLYNOMIAL
000403'054003-POLY3:   STA     3,SVAC3
000404'004475          JSR     POL2              ;0-2ND ORDER TERMS
000405'004534          JSR     POL3              ;3RD ORDER TERM
000406'004461          JSR     POLSG             ;SIGN
000407'004443          JSR     LIN9              ;ALTERNATE UNITS
000410'002006-         JMP     @C.ANS
                      ;AX/1024
000411'054455 LIN1:    STA     3,LINA3
000412'102000          ADC     0,0
000413'040004-         STA     0,POS
000414'030002-         LDA     2,SVAC2
000415'031004          LDA     2,4,2             ;A
000416'006004$         JSR     @MUL.U
000417'030411          LDA     2,D512
000420'147022          ADDZ    2,1,SZC
000421'101400          INC     0,0
000422'030407          LDA     2,D10M
```

```
000423'101220 LIN11: MOVZR   0,0
000424'125200        MOVR    1,1
000425'151404        INC     2,2,SZR
000426'000775        JMP     LIN11
000427'002437        JMP     @LINA3
000430'001000 D512:  512.
000431'177766 D10M:  -10.
              ; +C
000432'054434 LIN2:  STA     3,LINA3
000433'030002-       LDA     2,SVAC2
000434'031005        LDA     2,5,2           ;C
000435'147000        ADD     2,1
              ;CHECK SIGN
000436'102000        ADC     0,0             ;ASSUME +
000437'125113        MOVL#   1,1,SNC         ;CHECK FOR MINUS
000440'000403        JMP     .+3
000441'124400        NEG     1,1             ;NEGATE ANSWER
000442'102400        SUB     0,0
000443'040004-       STA     0,POS
000444'002422        JMP     @LINA3
              ; *C
                     .ENT    LIN3
000445'054421 LIN3:  STA     3,LINA3
000446'030002-       LDA     2,SVAC2
000447'031005        LDA     2,5,2           ; C
000450'006004$       JSR     @MUL.U
000451'002415        JMP     @LINA3
              ;ALTERNATE UNITS FOR +/- IF LOGGING
000452'054414 LIN9:  STA     3,LINA3
000453'034005-       LDA     3,IFLOG
000454'175005        MOV     3,3,SNR
000455'002411        JMP     @LINA3
000456'030002-       LDA     2,SVAC2
000457'035006        LDA     3,6,2           ;+ UNITS
000460'101005        MOV     0,0,SNR
000461'035007        LDA     3,7,2           ;- UNITS
000462'055001        STA     3,1,2
000463'102000        ADC     0,0
000464'040004-       STA     0,POS
000465'006401        JSR     @LINA3
              ;STORAGE
                     .ENT    POLSG,POL2,POL3,POLX,POLY       ;***DEBUG
000466'000000 LINA3: .-.
              ;DETERMINE SIGN OF POLY
000467'054777 POLSG: STA     3,LINA3
              ;
000470'024440        LDA     1,POLY
000471'102000        ADC     0,0             ;ASSUME +
000472'125113        MOVL#   1,1,SNC         ;CHECK FOR -
000473'000403        JMP     .+3
000474'124400        NEG     1,1
000475'102400        SUB     0,0
000476'040004-       STA     0,POS
000477'044431        STA     1,POLY
000500'002756        JMP     @LINA3
              ;SUM 0TH THRU 2ND ORDER TERMS
000501'054765 POL2:  STA     3,LINA3
              ;
000502'020726        LDA     0,D512
000503'106400        SUB     0,1
```

```
000504'044423         STA     1,POLX
                    ;
000505'030002-        LDA     2,SVAC2
000506'025005         LDA     1,5,2               ;1ST ORDER COEF
000507'030420         LDA     2,POLX
000510'006004$        JSR     @MUL.U
000511'030420         LDA     2,D5120
000512'006005$        JSR     @DIV.
                    ;
000513'030002-        LDA     2,SVAC2
000514'021004         LDA     0,4,2               ;ZERO ORDER COEF
000515'106400         SUB     0,1
000516'044412         STA     1,POLY
                    ;
000517'030002-        LDA     2,SVAC2
000520'025010         LDA     1,10,2              ;2ND ORDER COEF
000521'004411         JSR     PWER
000522'004410         JSR     PWER
000523'020405         LDA     0,POLY              ;GET FIRST TWO TERMS
000524'122400         SUB     1,0                 ;ADD 2ND ORDER TERM
000525'040403         STA     0,POLY              ;STORE SUM
000526'002740         JMP     @LINA3              ;RETURN
                    ;
000527'000000 POLX:   .-.
000530'000000 POLY:   .-.
000531'012000 D5120:  5120.
                    ;(AC1)*POLX/512
000532'054406 PWER:   STA      3,PWER3
000533'030774         LDA      2,POLX
000534'006004$        JSR      @MUL.U
000535'030673         LDA      2,D512
000536'006005$        JSR      @DIV.
000537'002401         JMP      @PWER3
000540'000000 PWER3:           .-.
                    ;ADD 3RD ORDER TERM TO POLY
000541'054725 POL3:   STA      3,LINA3
                    ;
000542'030002-        LDA      2,SVAC2
000543'025011         LDA      1,11,2             ;3RD ORDER COEF
000544'030763         LDA      2,POLX
000545'006004$        JSR      @MUL.U
000546'030761         LDA      2,POLX
000547'006004$        JSR      @MUL.U
000550'101400         INC      0,0
000551'105220         MOVZR    0,1
000552'004760         JSR      PWER
                    ;
000553'020755         LDA      0,POLY
000554'123000         ADD      1,0
000555'040753         STA      0,POLY
000556'002710         JMP      @LINA3
                    ;ARC SIN(AX/1024)
                      .ENT     SIN1
000557'054003-SIN1:   STA      3,SVAC3
000560'004405         JSR      GASIN
000561'002006-        JMP      @C.ANS
                    ;ARC COS(AX/1024)
                      .ENT     COS1
000562'054003-COS1:   STA      3,SVAC3
000563'004422         JSR      GACOS
```

```
 0009 PVAL
000564'002006-         JMP     @C.ANS
                ;ARC SIN
000565'054417 GASIN:   STA     3,GASA3
000566'004623          JSR     LIN1
000567'004643          JSR     LIN2
000570'020004-         LDA     0,POS
000571'040444          STA     0,SSIN
000572'034441          LDA     3,ATSIN
000573'021400 GAS10:   LDA     0,0,3
000574'106433          SUBZ#   0,1,SNC
000575'000403          JMP     GAS20            ;AC0>AC1
000576'175400          INC     3,3
000577'000774          JMP     GAS10
000600'030433 GAS20:   LDA     2,ATSIN
000601'156400          SUB     2,3
000602'165000          MOV     3,1
000603'002401          JMP     @GASA3
000604'000000 GASA3:   .-.
                ;ARC COS
000605'054425 GACOS:   STA     3,GACA3
000606'004603          JSR     LIN1
000607'004623          JSR     LIN2
000610'020004-         LDA     0,POS
000611'040425          STA     0,SCOS
000612'034421          LDA     3,ATSIN
000613'021400 GAC10:   LDA     0,0,3
000614'122432          SUBZ#   1,0,SZC
000615'000403          JMP     GAC20            ;AC0>=AC1
000616'175400          INC     3,3
000617'000774          JMP     GAC10
000620'030413 GAC20:   LDA     2,ATSIN
000621'172400          SUB     3,2
000622'024412          LDA     1,D90
000623'020413          LDA     0,SCOS
000624'101005          MOV     0,0,SNR
000625'146401          SUB     2,1,SKP
000626'147000          ADD     2,1
000627'152000          ADC     2,2
000630'050004-         STA     2,POS
000631'002401          JMP     @GACA3
000632'000000 GACA3:   .-.
                ;STORAGE
000633'000637'ATSIN:   TSIN    ;TABLE IF SINE VALUES
000634'000132 D90:     90.
000635'000000 SSIN:    .-.
000636'000000 SCOS:    .-.
                       .TITL   TSIN    ;TABLE OF SINE VALUES
                       .ENT    TSIN
000637'000011 TSIN:    009.
000640'000032          026.
000641'000054          044.
000642'000075          061.
000643'000116          078.
000644'000140          096.
000645'000161          113.
000646'000203          131.
000647'000224          148.
000650'000245          165.
000651'000266          182.    ;10 DEG
000652'000307          199.
```

```
 0010 PVAL
01 000653'000330         216.
02 000654'000351         233.
03 000655'000372         250.
04 000656'000413         267.
05 000657'000434         284.
06 000660'000455         301.
07 000661'000475         317.
08 000662'000516         334.
09 000663'000536         350.     ;20 DEG
10 000664'000557         367.
11 000665'000577         383.
12 000666'000617         399.
13 000667'000637         415.
14 000670'000657         431.
15 000671'000676         446.
16 000672'000716         462.
17 000673'000735         477.
18 000674'000754         492.
19 000675'000774         508.     ;30 DEG
20 000676'001012         522.
21 000677'001031         537.
22 000700'001050         552.
23 000701'001066         566.
24 000702'001105         581.
25 000703'001123         595.
26 000704'001141         609.
27 000705'001157         623.
28 000706'001174         636.
29 000707'001211         649.     ;40 DEG
30 000710'001227         663.
31 000711'001244         676.
32 000712'001260         688.
33 000713'001275         701.
34 000714'001311         713.
35 000715'001325         725.
36 000716'001341         737.
37 000717'001355         749.
38 000720'001370         760.
39 000721'001404         772.     ;50 DEG
40 000722'001417         783.
41 000723'001431         793.
42 000724'001444         804.
43 000725'001456         814.
44 000726'001470         824.
45 000727'001502         834.
46 000730'001513         843.
47 000731'001525         853.
48 000732'001536         862.
49 000733'001546         870.     ;60 DEG
50 000734'001557         879.
51 000735'001567         887.
52 000736'001577         895.
53 000737'001607         903.
54 000740'001616         910.
55 000741'001625         917.
56 000742'001634         924.
57 000743'001642         930.
58 000744'001651         937.
59 000745'001657         943.     ;70 DEG
60 000746'001664         948.
```

```
000747'001672       954.
000750'001677       959.
000751'001704       964.
000752'001710       968.
000753'001714       972.
000754'001720       976.
000755'001724       980.
000756'001727       983.
000757'001732       986.       ;80 DEG
000760'001735       989.
000761'001737       991.
000762'001742       994.
000763'001743       995.
000764'001745       997.
000765'001746       998.
000766'001747       999.
                    .ENT       SVOLT,SSTRP,SSTRM,STMP2,SACCL,SPRES,SRUD2,SPRO2
                    .ENT       SG177,SG183,SG186,SG190,SG194,SG195,SG196
              ;VOLTS
000767'003750"SVOLT: TE001*2
000770'003750"       TE001*2
000771'000306'       LINC
000772'000002        2
000773'003720        2000.
000774'176030        -1000.
              ;MICROSTRAIN (+)
000775'003764"SSTRP: TE011*2
000776'003764"       TE011*2
000777'000306'       LINC
001000'000004        4
001001'007640        4000.
001002'174060        -2000.
              ;MICROSTRAIN (-)
001003'004006"SSTRM: TE013*2
001004'004006"       TE013*2
001005'000306'       LINC
001006'100004        4+1B0
001007'007640        4000.
001010'174060        -2000.
              ;S/G GF=1.77
001011'003764"SG177: TE011*2
001012'003764"       TE011*2
001013'000306'       LINC
001014'000004        4
001015'006724        3540.
001016'174426        -1770.
              ;S/G  GF=1.83
001017'003764"SG183: TE011*2
001020'003764"       TE011*2
001021'000306'       LINC
001022'000004        4
001023'007114        3660.
001024'174332        -1830.
              ;S/G  GF=1.86
001025'003764"SG186: TE011*2
001026'003764"       TE011*2
001027'000306'       LINC
001030'000004        4
001031'007210        3720.
001032'174274        -1860.
```

```
0012 PVAL
01                         ;S/G  GF=1.90
02 001033'003764"SG190:  TE011*2
03 001034'003764"        TE011*2
04 001035'000306'        LINC
05 001036'000004         4
06 001037'007330         3800.
07 001040'174224         -1900.
08                         ;S/G  GF=1.94
09 001041'003764"SG194:  TE011*2
10 001042'003764"        TE011*2
11 001043'000306'        LINC
12 001044'007450         3880.
13 001045'174154         -1940.
14                         ;S/G  GF=1.95
15 001046'003764"SG195:  TE011*2
16 001047'003764"        TE011*2
17 001050'000306'        LINC
18 001051'000004         4
19 001052'007474         3900.
20 001053'174142         -1950.
21                         ;S/G  GF=1.96
22 001054'003764"SG196:  TE011*2
23 001055'003764"        TE011*2
24 001056'000306'        LINC
25 001057'000004         4
26 001060'007520         3920.
27 001061'174130         -1960.
28                         ;AIR TEMPERATURE THERMISTOR  231
29 001062'004060"STMP1:  TE021*2
30 001063'004060"        TE021*2
31 001064'000306'        LINC
32 001065'000004         4
33 001066'000310         200.
34 001067'177634         -100.
35                         ;THERMISTORS 5-10,171-172
36 001070'004060"STMP2:  TE021*2
37 001071'004060"        TE021*2
38 001072'000306'        LINC
39 001073'100003         3+1B0
40 001074'002171         1145.
41 001075'177230         -360.
42                         ;ACCEL  1,2,31,212-214
43 001076'004530"SACCL:  TE091*2
44 001077'004530"        TE091*2
45 001100'000306'        LINC
46 001101'000003         3
47 001102'000062         50.
48 001103'177747         -25.
49                         ;BUBBLER PRESSURE  187,188
50 001104'004634"SPRES:  TE121*2
51 001105'004634"        TE121*2
52 001106'000303'        LINP
53 001107'000004         4
54 001110'000530         344.
55                         ;RUDDER TORQUE 202
56 001111'004616"SRUD2:  TE120*2
57 001112'004616"        TE120*2
58 001113'000303'        LINP
59 001114'000002         2
60 001115'000530         344.
```

```
 0013 PVAL
              ;PROP DIFFERENTIAL PRESS 203
001116'004616"SPRO2: TE120*2
001117'004616"       TE120*2
001120'000303'       LINP
001121'000003        3
001122'000212        138.
                     .ENT    STRQ2,SRPM,SRPMS,STHR,SPCT
                     .ENT    SRUDR,SPRO1,SVISI,STENT,SFUEL
              ;SHAFT TORQUE  208
001123'004724"STRQ2: TE125*2
001124'004724"       TE125*2
001125'000303'       LINP
001126'000004        4
001127'001013        523.
              ;ENGINE RPM  241
001130'004470"SRPM:  TE062*2
001131'004470"       TE062*2
001132'000306'       LINC
001133'000004        4
001134'001070        568.
001135'177774        -4.
              ;SHAFT RPM  197
001136'004470"SRPMS: TE062*2
001137'004470"       TE062*2
001140'000306'       LINC
001141'000004        4
001142'000454        300.
001143'177552        -150.
              ;THRUST  198
001144'004272"STHR:  TE031*2
001145'004272"       TE031*2
001146'000306'       LINC
001147'000004        4
001150'000360        240.
001151'177610        -120.
              ;FUEL RACK  199
001152'004476"SFUEL: TE071*2
001153'004476"       TE071*2
001154'000306'       LINC
001155'000004        4
001156'000375        253.
001157'177745        -27.
                     ;RUDDER ANGLE RESPONSE  200
001160'004200"SRUD1: TE026*2
001161'000000        .-.
001162'000312'       LINU
001163'000003        3
001164'001332        730.
001165'177223        -365.
001166'004252"       TE028*2
001167'004240"       TE027*2
              ;PROP PITCH RESP  201,244
001170'004114"SPRO1: TE023*2
001171'000000        .-.     ;DEPENDS ON SIGN
001172'000312'       LINU
001173'000004        4
001174'000060        48.
001175'177752        -22.
001176'004154"       TE024*2
001177'004166"       TE025*2
```

```
            ;BUBBLER MIMIC 204,205,206
                     .ENT    SBUBL
001200'004606"SBUBL: TE113*2
001201'004606"       TE113*2
001202'000323'       BUBBL
            ;VISIBILITY  224
001203'004550"SVISI: TE101*2
001204'004550"       TE101*2
001205'000303'       LINP
001206'000004        4
001207'000012        10.
            ;CLOUD & ICE COVER  225,226
001210'004576"STENT: TE111*2
001211'004576"       TE111*2
001212'000303'       LINP
001213'000004        4
001214'000012        10.
                     .ENT    SBAR1,SWIND,SWDIR,SLOG,SSHIP
                     .ENT    SSINE,SCOSE,SDRAFT
            ;BAROMETRIC PRESSURE  228,229,230
001215'004634"SBAR1: TE121*2
001216'004646"       TE121*2+12
001217'000303'       LINP
001220'000004        4
001221'000011        9.
001222'004652"SBAR2: TE122*2
001223'004652"       TE122*2
001224'000303'       LINP
001225'000004        4
001226'000012        10.
001227'004670"SBAR3: TE123*2
001230'004670"       TE123*2
001231'000303'       LINP
001232'000004        4
001233'000012        10.
001234'004706"SBAR4: TE124*2
001235'004706"       TE124*2
001236'000303'       LINP
001237'000004        4
001240'000012        10.
            ;PERCENT  232
001241'004476"SPCT:  TE071*2
001242'004476"       TE071*2
001243'000303'       LINP
001244'000004        4
001245'000144        100.
            ;WIND SPEED  233
001246'004514"SWIND: TE081*2
001247'004514"       TE081*2
001250'000303'       LINP
001251'000003        3
001252'000213        139.
            ;WIND DIR  234
001253'004046"SWDIR: TE020*2
001254'004046"       TE020*2
001255'000306'       LINC
001256'100004        4+1B0
001257'000550        360.
001260'177330        -296.
            ;LOG 235
```

```
0015 PVAL
       000767'SLOG=  SVOLT
              ;SHIP SPEED 236
001261'004514"SSHIP: TE081*2
001262'004514"       TE081*2
001263'000303'       LINP
001264'000003        3
001265'000464        308.
              ;SINE  237
001266'004370"SSINE: TE041*2
001267'004370"       TE041*2
001270'000557'       SIN1
001271'000004        4
001272'003720        2000.
001273'176030        -1000.
              ;COSINE  238
001274'004424"SCOSE: TE044*2
001275'004424"       TE044*2
001276'000562'       COS1
001277'000004        4
001300'003720        2000.
001301'176030        -1000.
                     .ENT    SPTCH,SROLL,SPROP,S10PC,STU
              ;PITCH  239
001302'004312"SPTCH: TE037*2
001303'000000        .-.
001304'000312'       LINU
001305'000004        4
001306'000132        90.
001307'177723        -45.
001310'004344"       TE038*2
001311'004356"       TE039*2
              ;ROLL  240
001312'004200"SROLL: TE026*2
001313'000000        .-.
001314'000312'       LINU
001315'000004        4
001316'000170        120.
001317'177704        -60.
001320'004240"       TE027*2
001321'004252"       TE028*2
              ;DRAFT
001322'004750"SDRAFT:        TE131*2
001323'004750" -     TE131*2
001324'000306'       LINC
001325'000004        4
001326'000005        5.
001327'000007        7.
              ;PROP PITCH SET  244
001330'004114"SPROP: TE023*2
001331'000000        .-.      ;DEPENDS ON SIGN
001332'000373'       POLY2
001333'000003        3
001334'000152        106.
001335'010266        4278.
001336'004154"       TE024*2
001337'004166"       TE025*2
001340'000335        221.
              ;RUDDER ANGLE COMMAND  245
001341'004200"SRUDR: TE026*2
001342'000000        .-.
```

```
 0016 PVAL
01 001343'000312'         LINU
02 001344'000003          3
03 001345'001456          814.
04 001346'177151          -407.
05 001347'004252"         TE028*2
06 001350'004240"         TE027*2
07                  ;PERCENT (WINDOWS)
08 001351'004476"S10PC:   TE071*2
09 001352'004510"         TE072*2
10 001353'000317'         LINW
11 001354'000004          4
12 001355'000012          10.
13 001356'000012          10.
14                  ;SPECIAL OPTION
15 001357'004762"STU:     TE991*2
16 001360'005016"         TE992*2
17 001361'000303'         LINP
18 001362'000004          4
19 001363'002000          1024.
20                  ;REPETITION MACRO
21                          .MACRO  REPT
22                          .DO     ^1
23                          ^2
24                          .ENDC
25                 %
26        000001            .NOMAC  1
                            .TITL   STAB
28                          .ENT    STAB
29 001364'000767'STAB:      SVOLT                      ;0
30                          REPT    2 SACCL            ;1-2
31 001367'000775'           SSTRP                      ;3-4
32 001370'000775'           SSTRP
33                          REPT    6 STMP2            ;5-10
34                          REPT    3        SACCL     ;11-13 ACCEL FILTERS
35                          REPT    3        SG195     ;14-16
36                          REPT    3        SG190     ;17-19
37 001410'001046'           SG195                      ;20
38                          REPT    2        SG190     ;21-22
39                          REPT    8.       SG195     ;23-30
40 001423'001076'           SACCL                      ;31
41                   ;BOX 2,3,4
42 001424'001033'           SG190                      ;32
43 001425'001017'           SG183                      ;33
44                          REPT    2        SG177     ;34-35
45 001430'001046'           SG195                      ;36
46 001431'001017'           SG183                      ;37
47 001432'001033'           SG190                      ;38
48                          REPT    2        SG195     ;39-40
49                          REPT    2        SG196     ;41-42
50 001437'001011'           SG177                      ;43
51                          REPT    4        SG183     ;44-47
52 001444'001011'           SG177                      ;48
53                          REPT    14.      SG183     ;49-62
54                          REPT    3        SG195     ;63-65
55                          REPT    5        SG183     ;66-70
56 001473'001033'           SG190                      ;71
57                          REPT    2        SG195     ;72-73
58                          REPT    2        SG183     ;74-75
59                          REPT    2        SG186     ;76-77
60                          REPT    2        SG195     ;78-79
```

```
0017 PVAL
                           REPT     5       SG183    ;80-84
001511'001054'             SG196                     ;85
                           REPT     4       SG183    ;86-89
001516'001054'             SG196                     ;90
                           REPT     15.     SG183    ;91-105
                           REPT     4       SG196    ;106-109
                           REPT     8.      SG183    ;110-117
                           REPT     2       SG196    ;118-119
                           REPT     6       SG183    ;120-125
001562'001046'             SG195                     ;126
                           REPT     2       SG183    ;127-128
                           REPT     2       SG177    ;129-130
                           REPT     2       SG183    ;131-132
                           REPT     3       SG196    ;133-135
001574'001017'             SG183                     ;136
001575'001054'             SG196                     ;137
001576'001017'             SG183                     ;138
                           REPT     8.      SG196    ;139-146
                           REPT     2       SG183    ;147-148
                           REPT     6       SG196    ;149-154
                           REPT     2       SG177    ;155-156
                           REPT     5       SG183    ;157-161
001626'001041'             SG194                     ;162
                           REPT     3       SG183    ;163-165
                           REPT     3       SG196    ;166-168
001635'001017'             SG183                     ;169
                           REPT     4       SG196    ;170-173
                           REPT     13.     SG177    ;174-186
                    ;BOX 5
                           REPT       2. SPRES       ;187-188
                           REPT       9. SVOLT       ;189-197
                    ;BOX 6
001672'001144'             STHR     ;198 THRUST
001673'001152'             SFUEL    ;199 FUEL RACK SETTING
001674'001160'             SRUD1    ;200 RUDDER RESPONSE
001675'001170'             SPRO1    ;201 PROP RESPONSE
001676'001111'             SRUD2    ;202 RUDDER TORQUE
001677'001116'             SPRO2    ;203 PROP DIFFERENTIAL PRESSURE
                           REPT       3. SBUBL       ;204-206 BUBL MIMIC
001703'000767'             SVOLT                     ;207
001704'001123'             STRQ2                     ;208
                           REPT     3        SVOLT   ;209-211
                           REPT       3. SACCL       ;212-214 ACCELEROMETERS
                           REPT     3        SACCL   ;215-217 ACCEL FILTERS
                           REPT       6. SVOLT       ;215-223
                    ;BOX 7
001724'001203'             SVISI    ;224 VISIBILITY
001725'001210'             STENT    ;225 CLOUD COVER
001726'001210'             STENT    ;226 ICE COVER
001727'000767'             SVOLT    ;227 BAROMETRIC PRESSURE (DIGIT 1)
001730'001222'             SBAR2    ;228   (DIGIT 2)
001731'001227'             SBAR3    ;229   (DIGIT 3)
001732'001234'             SBAR4    ;230   (DIGIT 4)
001733'001062'             STMP1    ;231 AIR TEMP
001734'001241'             SPCT     ;232 REL HUMIDITY
001735'001246'             SWIND    ;233 WIND VEL
001736'001253'             SWDIR    ;234 WIND DIR
001737'000767'             SVOLT    ;235 LOG
001740'001261'             SSHIP    ;236 SHIP SPEED
001741'001266'             SSINE    ;237 SHIP HEAD (SINE)
```

```
0018 PVAL
001742'001274'          SCOSE   ;238 SHIP HEAD (COSINE)
001743'001302'          SPTCH   ;239 PITCH ANGLE
001744'001312'          SROLL   ;240 ROLL ANGLE
001745'001130'          SRPM    ;241 ENGINE RPM
                        REPT     2. SVOLT       ;242-243
001750'001330'          SPROP                   ;244 PROP COMMAND
001751'001341'          SRUDR                   ;245 RUDDER COMMAND
                        REPT     6. SVOLT       ;246-251
                        REPT     2. SDRAFT      ;252-253 DRAFT
                        REPT     2. SVOLT       ;254-255
                        .TITL   TEXT
                        .ENT    TE001
       000001           .TXTM   1
001764'053117 TE001:    .TXT    /VOLTS<15>/
       046124
       051415
       000000
001770'053015 TE002:    .TXT    /V<15>/
       000000
001772'046511 TE011:    .TXT    /MICROSTRAINS<15>/
       041522
       047523
       052122
       040511
       047123
       006400
002001'072523 TE012:    "U+40*400+"S
002002'006400           15*400
002003'046511 TE013:    .TXT    /MICROSTRAINS (INVERTED)<15>/
       041522
       047523
       052122
       040511
       047123
       020050
       044516
       053105
       051124
       042504
       024415
       000000
002020'072523 TE014:    "U+40*400+"S
002021'020052           .TXT    / *<15>/
       006400
002023'042105 TE020:    .TXT    /DEGREES<15>/
       043522
       042505
       051415
       000000
002030'042105 TE021:    .TXT    /DEGREES CENTIGRADE<15>/
       043522
       042505
       051440
       041505
       047124
       044507
       051101
       042105
       006400
002042'042105 TE022:    .TXT    /DEG C<15>/
```

```
 0019 PVAL
              043440
              041415
              000000
   002046'042111  TE023: .TXT    /DIVISIONS (+ AHEAD, - ASTERN)<15>/
              053111
              051511
              047516
              051440
              024053
              020101
              044105
              040504
              026040
              026440
              040523
              052105
              051116
              024415
              000000
   002066'042111  TE024: .TXT    /DIV AHD<15>/
              053040
              040510
              042015
              000000
   002073'042111  TE025: .TXT    /DIV AST<15>/
              053040
              040523
              052015
              000000
   002100'042105  TE026: .TXT    /DEGREES (+ PORT, - STARBOARD)<15>/
              043522
              042505
              051440
              024053
              020120
              047522
              052054
              020055
              020123
              052101
              051102
              047501
              051104
              024415
              000000
   002120'042105  TE027: .TXT    /DEG PORT<15>/
              043440
              050117
              051124
              006400
   002125'042105  TE028: .TXT    /DEG STBD<15>/
              043440
              051524
              041104
              006400
   002132'042105  TE029: .TXT    /DEG<15>/
              043415
              000000
   002135'052117  TE031: .TXT    /TONNES<15>/
              047116
 0020 PVAL
01          042523
02          006400
03 002141'046520 TE032: .TXT    /MPONS<15>/
04          047516
05          051415
06          000000
07 002145'042105 TE037: .TXT    /DEGREES (+ FORE, - AFT)<15>/
08          043522
09          042505
10          051440
11          024053
12          020106
13          047522
14          042454
15          020055
16          020101
17          043124
18          024415
19          000000
20 002162'042105 TE038: .TXT    /DEG FORE<15>/
21          043440
22          043117
23          051105
24          006400
25 002167'042105 TE039: .TXT    /DEG AFT<15>/
26          043440
27          040506
28          052015
29          000000
30 002174'042105 TE041: .TXT    /DEGREES (SINE)<15>/
31          043522
32          042505
33          051440
34          024123
35          044516
36          042451
37          006400
38 002204'042105 TE042: .TXT    /DEG (SIN)<15>/
39          043440
40          024123
41          044516
42          024415
43          000000
44 002212'042105 TE044: .TXT    /DEGREES (COSINE)<15>/
45          043522
46          042505
47          051440
48          024103
49          047523
50          044516
51          042451
52          006400
53 002223'042105 TE045: .TXT    /DEG (COS)<15>/
54          043440
55          024103
56          047523
57          024415
58          000000
59 002231'050123 TE052: .TXT    /PSI<15>/
60          044415
```

```
0021 PVAL
           000000
   002234'051120 TE062: .TXT    /RPM<15>/
           046415
           000000
   002237'050105 TE071: .TXT    /PERCENT<15>/
           051103
           042516
           052015
           000000
   002244'022415 TE072: .TXT    /%<15>/
           000000
   002246'045516 TE081: .TXT    /KNOTS<15>/
           047524
           051415
           000000
   002252'045516 TE082: .TXT    /KN<15>/
           006400
   002254'043415 TE091: .TXT    /G<15>/
           000000
   002256'043522 TE092: .TXT    /GRAVITIES<15>/
           040526
           044524
           044505
           051415
           000000
   002264'047101 TE101: .TXT    /NAUTICAL MILES<15>/
           052524
           044503
           040514
           020115
           044514
           042523
           006400
   002274'047040 TE102: .TXT    /N MI<15>/
           046511
           006400
   002277'052105 TE111: .TXT    /TENTHS<15>/
           047124
           044123
           006400
   002303'047516 TE113: .TXT    /ON-OFF<15>/
           026517
           043106
           006400
   002307'046505 TE120: .TXT    /MEGAPASCALS <15>/
           043501
           050101
           051503
           040514
           051440
           006400
   002316'045511 TE121: .TXT    /KILOPASCALS <15>/
           046117
           050101
           051503
           040514
           051440
           006400
   002325'046511 TE122: .TXT    /MILLIBARS B<15>/
           046114
```

```
 0022 PVAL
01          044502
02          040522
03          051440
04          041015
05          000000
06 002334'046511 TE123: .TXT    /MILLIBARS A<15>/
07          046114
08          044502
09          040522
10          051440
11          040415
12          000000
13 002343'046511 TE124: .TXT    /MILLIBARS R<15>/
14          046114
15          044502
16          040522
17          051440
18          051015
19          000000
20 002352'045511 TE125: .TXT    /KILONEWTON-METRES<15>/
21          046117
22          047105
23          053524
24          047516
25          026515
26          042524
27          051105
28          051415
29          000000
30 002364'046505 TE131: .TXT    /METRES <15>/
31          052122
32          042523
33          020015
34          000000
35 002371'052111 TE991: .TXT    /TIN PLATED SQUARE RABBITS<15>/
36          047040
37          050114
38          040524
39          042504
40          020123
41          050525
42          040522
43          042440
44          051101
45          041102
46          044524
47          051415
48          000000
49 002407'052123 TE992: .TXT    /TSPR<15>/
50          050122
51          006400
                        .TITL   PVAL
53                      .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
     0023 PVAL

    ADTEN 000214'         2/45     3/09     3/35     4/04
    APTEN 000215'         2/44     4/05
    ATAD  000065'         1/58     2/24
    ATSIN 000633'         9/08     9/14     9/25     9/31     9/43
    AVOLT 000066'         1/51     2/25
    BITS  000067'         1/23     2/26
    BUBBL 000323' EN      5/43     5/44    14/05
    C100  000367'         6/09     6/28
    C1740 000372'         5/48     6/31
    C20   000365'         5/46     6/26
    C200  000370'         6/03     6/29
    C40   000366'         6/15     6/27
    C400  000371'         5/57     6/30
    CAN10 000126'         2/56     3/02
    CAN30 000135'         3/05     3/09     3/36
    CAN40 000142'         3/14     3/16
    CAN50 000153'         3/20     3/24
    CAN60 000157'         3/22     3/28
    CAN70 000164'         3/29     3/35
    CAN80 000166'         3/12     3/39
    CAN85 000174'         3/42     3/45
    CAN90 000202'         3/47     3/51
    CANS  000102' EN      2/37     2/38     2/40     2/42
    COS1  000562' EN      8/58     8/59    15/18
    C.ANS 000006- EN      2/38     2/40     5/23     5/29     5/36     5/41     6/42
                          6/49     8/56     9/01
    D10M  000431'         6/60     7/07
    D512  000430'         6/57     7/06     7/59     8/31
    D5120 000531'         8/07     8/26
    D90   000634'         9/33     9/44
    DIV.  000005$ XD      1/03     8/08     8/32
    FIR10 000245'         4/26     4/34
    FIRA0 000251'         4/28     4/32     4/39
    FIRA3 000250'         4/19     4/22     4/36     4/38
    FIRST 000231' EN      3/06     3/24     3/31     4/17     4/18     4/19
    GAC10 000613'         9/26     9/30
    GAC20 000620'         9/28     9/31
    GACA3 000632'         9/20     9/40     9/41
    GACOS 000605'         8/60     9/20
    GAS10 000573'         9/09     9/13
    GAS20 000600'         9/11     9/14
    GASA3 000604'         9/03     9/17     9/18
    GASIN 000565'         8/55     9/03
    IFLOG 000005- EN      1/06     1/11     2/04     7/30
    IFRST 000207'         2/52     3/18     3/59     4/20     4/35
    IFTHR 000206' EN      3/45     3/57     3/58
    IFVLT 000001- EN      1/06     1/07     1/48
    LIN1  000411'         5/22     5/27     5/33     5/39     6/51     9/04     9/21
    LIN11 000423'         7/01     7/04
    LIN2  000432'         5/28     5/34     7/09     9/05     9/22
    LIN3  000445' EN      5/40     7/22     7/23
    LIN9  000452'         5/35     6/41     6/48     7/29
    LINA3 000466'         6/51     7/05     7/09     7/20     7/23     7/27     7/29
                          7/32     7/40     7/43     7/45     7/55     7/57     8/22
                          8/36     8/51
    LINC  000306' EN      5/25     5/26    11/22    11/29    11/36    11/43    11/50
                         11/57    12/04    12/11    12/17    12/24    12/31    12/38
                         12/45    13/18    13/25    13/32    13/39    14/56    15/44
    LINP  000303' EN      5/20     5/21    12/52    12/58    13/04    13/12    14/09
```

```
0024 PVAL

                        14/15   14/23   14/28   14/33   14/38   14/44   14/50
                        15/05   16/17
LINU   000312' EN        5/31    5/32   13/46   13/55   15/26   15/35   16/01
LINW   000317'           5/38   16/10
MUL.U  000004$ XD        1/03    6/56    7/26    8/06    8/30    8/41    8/43
NDIG   000210'           3/03    3/28    3/60
POL2   000501' EN        6/33    6/36    6/45    7/42    7/57
POL3   000541' EN        6/33    6/46    7/42    8/36
POLSG  000467' EN        6/33    6/37    6/47    7/42    7/45
POLX   000527' EN        6/33    7/42    8/01    8/05    8/24    8/29    8/40
                         8/42
POLY   000530' EN        6/33    7/42    7/47    7/54    8/13    8/19    8/21
                         8/25    8/48    8/50
POLY2  000373' EN        6/32    6/35   15/51
POLY3  000403' EN        6/32    6/44
POS    000004-           1/10    2/57    3/01    4/24    6/53    7/19    7/39
                         7/53    9/06    9/23    9/39
PTEN   000224'           4/05    4/12
PVA10  000012'           1/25    1/30
PVA20  000024'           1/34    1/42
PVA30  000026'           1/40    1/45
PVA40  000027'           1/28    1/43    1/46
PVA50  000037' EN        1/50    1/56    1/57
PVA80  000053'           1/54    2/06    2/12
PVA90  000055'           2/15
PVAA0  000061'           1/21    2/15    2/20
PVAA1  000062'           1/20    2/16    2/21
PVAA2  000063'           1/19    1/57    2/17    2/22
PVAA3  000064'           1/18    2/18    2/23
PVAL   000000' EN        1/01    1/02    1/05    1/18   22/52
PWER   000532'           8/17    8/18    8/28    8/46
PWER3  000540'           8/28    8/33    8/34
P.NL   000003$ XD        1/03    2/13    3/52    6/23
P.VAL  000000- EN        1/02    1/05
QANS   000075'           1/46    2/32    2/47    2/48    2/50    3/39    3/44
                         4/02    4/03
QB0DC  000217'           2/49    3/43    4/07
QBLBL  000074'           1/45    2/31    2/46
QBPT   000211'           2/43    3/21    3/49    4/01    4/29    4/52    5/02
                         5/52
QBPT0  000212'           3/51    4/02    5/51    6/22
QBPT1  000213'           2/42    4/03
QCR    000071'           2/27    2/28
QCRB   000070'           2/12    2/27
QDEC   000220'           3/07    3/32    4/08
QLMK   000302'           4/55    5/13
QMIDC  000216'           3/40    4/06
QMIN   000221'           4/09    4/30
QR0    000222'           3/25    4/10    5/56    6/02    6/08    6/14
QRBL   000223'           3/48    4/11    5/53    6/20
QRMK   000301'           4/49    5/12
QSTBL  000073'           1/42    2/30
QSTST  000072'           1/27    2/29
REPT   000000  MC       16/21   16/30   16/33   16/34   16/35   16/36   16/38
                        16/39   16/44   16/48   16/49   16/51   16/53   16/54
                        16/55   15/57   16/58   16/59   16/60   17/01   17/03
                        17/05   17/06   17/07   17/08   17/09   17/11   17/12
                        17/13   17/14   17/18   17/19   17/20   17/21   17/22
```

```
   0025 PVAL
                          17/24   17/25   17/27   17/28   17/30   17/31   17/39
                          17/42   17/43   17/44   17/45   18/05   18/08   18/09
                          18/10
  S10PC 001351' EN        15/22   16/08
  SACCL 001076' EN        11/17   12/43   16/31   16/35   16/40   17/44   17/45
  SBAR1 001215' EN        14/18   14/21
  SBAR2 001222'           14/26   17/51
  SBAR3 001227'           14/31   17/52
  SBAR4 001234'           14/36   17/53
  SBUBL 001200' EN        14/02   14/03   17/40
  SCOS  000636'            9/24    9/34    9/46
  SCOSE 001274' EN        14/19   15/16   18/01
  SDRAF 001322' EN        14/19   15/42   18/10
  SFUEL 001152' EN        13/08   13/37   17/34
  SG177 001011' EN        11/18   11/41   16/45   16/50   16/52   17/13   17/22
                          17/29
  SG183 001017' EN        11/18   11/48   16/43   16/46   16/52   16/54   16/56
                          16/59   17/02   17/04   17/06   17/09   17/10   17/12
                          17/14   17/15   17/17   17/20   17/23   17/25   17/26
  SG186 001025' EN        11/18   11/55   16/60
  SG190 001033' EN        11/18   12/02   16/37   16/39   16/42   16/47   16/56
  SG194 001041' EN        11/18   12/09   17/23
  SG195 001046' EN        11/18   12/15   16/36   16/37   16/40   16/45   16/49
                          16/55   16/58   17/01   17/10
  SG196 001054' EN        11/18   12/22   16/50   17/02   17/04   17/07   17/09
                          17/15   17/16   17/19   17/21   17/26   17/28
  SIN1  000557' EN         8/53    8/54   15/11
  SLOG  000767' EN        14/18   15/01
  SPCT  001241' EN        13/07   14/42   17/55
  SPRES 001104' EN        11/17   12/50   17/31
  SPRO1 001170' EN        13/08   13/53   17/36
  SPRO2 001116' EN        11/17   13/02   17/38
  SPROP 001330' EN        15/22   15/49   18/06
  SPTCH 001302' EN        15/22   15/24   18/02
  SROLL 001312' EN        15/22   15/33   18/03
  SRPM  001130' EN        13/07   13/16   18/04
  SRPMS 001136' EN        13/07   13/23
  SRUD2 001111' EN        11/17   12/56   17/37
  SRUDR 001341' EN        13/08   15/59   18/07
  SSHIP 001261' EN        14/18   15/03   17/59
  SSIN  000635'            9/07    9/45
  SSINE 001266' EN        14/19   15/09   17/60
  SSTRM 001003' EN        11/17   11/34
  SSTRP 000775' EN        11/17   11/27   16/31   16/32
  STAB  001364' EN         2/24   16/27   16/28   16/29
  STBA0 000275'            4/48    5/03    5/08
  STBA1 000276'            4/47    5/04    5/09
  STBA2 000277'            4/46    5/05    5/10
  STBA3 030300'            4/45    5/06    5/11
  STBT  000252' EN         3/08    3/27    3/33    3/50    4/31    4/40    4/41
                           4/45    5/54    5/60    6/06    6/12    6/18    6/21
  STENT 001210' EN        13/08   14/13   17/48   17/49
  STHR  001144' EN        13/07   13/30   17/33
  STMP1 001062'           12/29   17/54
  STMP2 001070' EN        11/17   12/36   16/34
  STRQ2 001123' EN        13/07   13/10   17/41
  STU   001357' EN        15/22   16/15
  SVAC2 000002-            1/08    1/52    2/01    2/08    2/53    6/38    6/54
```

```
0026 PVAL

                          7/10     7/24     7/33     8/03     8/10     8/15     8/38
$VAC3 000003-             1/09     3/54     5/21     5/26     5/32     5/38     5/44
                          6/24     6/35     6/44     8/54     8/59
$VISI 001203' EN         13/03    14/07    17/47
$VOLT 000767' EN          2/25    11/17    11/20    15/01    16/29    17/32    17/40
                         17/43    17/46    17/50    17/58    18/06    18/09    18/11
$WDIR 001253' EN         14/18    14/54    17/57
$WIND 001246' EN         14/19    14/48    17/56
TAB3A 000001$ XD          1/03     1/30
TAB4A 000002$ XD          1/03     1/36
TE001 001764' EN         11/20    11/21    18/12    18/14
TE002 001770'            18/18
TE011 001772'            11/27    11/28    11/41    11/42    11/48    11/49    11/55
                         11/56    12/02    12/03    12/09    12/10    12/15    12/16
                         12/22    12/23    18/20
TE012 002001'            18/27
TE013 002003'            11/34    11/35    18/29
TE014 002020'            18/42
TE020 002023'            14/54    14/55    18/45
TE021 002030'            12/29    12/30    12/36    12/37    18/50
TE022 002042'            18/60
TE023 002046'            13/53    15/49    19/04
TE024 002066'            13/59    15/55    19/20
TE025 002073'            13/60    15/56    19/25
TE026 002100'            13/44    15/33    15/59    19/30
TE027 002120'            13/51    15/39    16/06    19/46
TE028 002125'            13/50    15/40    16/05    19/51
TE029 002132'            19/56
TE031 002135'            13/30    13/31    19/59
TE032 002141'            20/03
TE037 002145'            15/24    20/07
TE038 002162'            15/30    20/20
TE039 002167'            15/31    20/25
TE041 002174'            15/09    15/10    20/30
TE042 002204'            20/38
TE044 002212'            15/16    15/17    20/44
TE045 002223'            20/53
TE052 002231'            20/59
TE062 002234'            13/16    13/17    13/23    13/24    21/02
TE071 002237'            13/37    13/38    14/42    14/43    16/08    21/05
TE072 002244'            16/09    21/10
TE081 002246'            14/48    14/49    15/03    15/04    21/12
TE082 002252'            21/16
TE091 002254'            12/43    12/44    21/18
TE092 002256'            21/20
TE101 002264'            14/07    14/08    21/26
TE102 002274'            21/34
TE111 002277'            14/13    14/14    21/37
TE113 002303'            14/03    14/04    21/41
TE120 002307'            12/56    12/57    13/02    13/03    21/45
TE121 002316'            12/50    12/51    14/21    14/22    21/52
TE122 002325'            14/26    14/27    21/59
TE123 002334'            14/31    14/32    22/06
TE124 002343'            14/36    14/37    22/13
TE125 002352'            13/10    13/11    22/20
TE131 002364'            15/42    15/43    22/30
TE991 002371'            16/15    22/35
TE992 002407'            16/16    22/49
```

```
 0027 PVAL
TSIN  000637' EN    9/43    9/47    9/48    9/49
```

```
0001 DT000 MACRO REV 06.50                  19:12:14 10/31/80
                               .TITL   DT000   ;DATE & TIME
                               .EXTN   VB100,Q13.2
                               .EXTD   IBUF,DIG.,P.CR,R.LN,GT.CH.BN.AS
                               .EXTD   CK.CR
                     ;
                               .ZREL
                               .ENT    DTYR,DTMO,DTDA,DTHR,DTMI,DTSC
   000000-000000 DTYR:   .-.
   000001-000000 DTMO:   .-.
   000002-000000 DTDA:   .-.
   000003-000000 DTHR:   .-.
   000004-000000 DTMI:   .-.
   000005-000000 DTSC:   .-.
                     ;
                               .NREL
          000001               .TXTM   1
                               .ENT    DT001
                 DT001:
   000000'005040 Q00:    .TXT    /<12> * * * DATE & TIME * * *<15>/
          025040
          025040
          025040
          042101
          052105
          020046
          020124
          044515
          042440
          025040
          025040
          025015
          000000
   000016'042101 Q01:    .TXT    ]DATE (Y,M,D): ]
          052105
          020050
          054454
          046454
          042051
          035040
          000000
   000026'052111 Q02:    .TXT    ]TIME (H,M,S): ]
          046505
          020050
          044054
          046454
          051451
          035040
          000000
   000036'053101 Q06:    .TXT    /VALUE ERROR<15>/
          046125
          042440
          042522
          051117
          051015
          000000
                     ;
   000045'000000"Q00B:   Q00*2
   000046'000034"Q01B:   Q01*2
   000047'000054"Q02B:   Q02*2
   000050'177777 VBSUB:  VB100
```

```
                      .TITL   DT000
                      .ENT    DT000
               ;
               ;   GET PRESENT DATE
               ;
000051'040534  DT000:  STA     0,DT0A0
000052'044534          STA     1,DT0A1
000053'050534          STA     2,DT0A2
000054'054534          STA     3,DT0A3
               ;
000055'020770          LDA     0.Q00B          ;SKIP A LINE
000056'006003$         JSR     @P.CR
000057'006771          JSR     @VBSUB
000060'102400          SUB     0,0
000061'040530          STA     0,DTCNG
               ;   GET DATE
000062'006017          .SYSTEM
000063'021006          .GDAY
000064'000401          JMP     .+1
000065'151004          MOV     2,2,SZR
               ;TYPE CURRENT
000066'004541          JSR     DT100
               ;ASK FOR DATE
000067'020757  DT010:  LDA     0,Q01B          ;ASK FOR DATE
000070'006003$         JSR     @P.CR
000071'006004$         JSR     @R.LN           ;READ RESPONSE
000072'000437          JMP     DT020
               ;
000073'020001$         LDA     0,IBUF
000074'006002$         JSR     @DIG.           ;DECODE YEAR
000075'044000-         STA     1,DTYR
000076'151005          MOV     2,2,SNR
000077'000514          JMP     DT050
               ;
000100'006005$         JSR     @GT.CH          ;SKIP DELIMITER
000101'030511          LDA     2,QRCR
000102'132405          SUB     1,2,SNR
000103'000510          JMP     DT050           ;IF CR, FORMAT ERROR
000104'006002$         JSR     @DIG.           ;DECODE MONTH
000105'044001-         STA     1,DTMO
000106'151005          MOV     2,2,SNR
000107'000504          JMP     DT050
               ;
000110'006005$         JSR     @GT.CH          ;SKIP DELIMITER
000111'030501          LDA     2,QRCR
000112'132405          SUB     1,2,SNR
000113'000500          JMP     DT050           ;IF CR, FORMAT ERROR
000114'006002$         JSR     @DIG.           ;DECODE DAY
000115'044002-         STA     1,DTDA
000116'151005          MOV     2,2,SNR
000117'000474          JMP     DT050
               ;
000120'006007$         JSR     @CK.CR
000121'000746          JMP     DT010
               ;SET DATE
000122'020002-         LDA     0,DTDA
000123'024001-         LDA     1,DTMO
000124'030000-         LDA     2,DTYR
000125'006017          .SYSTEM
000126'021005          .SDAY
```

```
0003 DT000
000127'000466          JMP     DT060
000130'040461          STA     0,DTCNG
                ;ASK FOR TIME
                       .ENT    DT020
000131'020716 DT020:   LDA     0,Q02B                   ;ASK FOR TIME
000132'006003$         JSR     @P.CR
000133'006004$         JSR     @R.LN                    ;GET RESPONSE
000134'000442          JMP     DT030                    ;JUST CR
                ;
000135'102400          SUB     0,0
000136'040005-         STA     0,DTSC
000137'040004-         STA     0,DTMI
000140'020001$         LDA     0,IBUF
000141'006002$         JSR     @DIG.                    ;DECODE HOUR
000142'044003-         STA     1,DTHR
000143'151005          MOV     2,2,SNR
000144'000454          JMP     DT070
                ;
000145'006005$         JSR     @GT.CH                   ;SKIP DELIMITER
000146'030444          LDA     2,QRCR
000147'146405          SUB     2,1,SNR
000150'000417          JMP     DT026
000151'006002$         JSR     @DIG.                    ;DECODE MINUTE
000152'044004-         STA     1,DTMI
000153'151005          MOV     2,2,SNR
000154'000444          JMP     DT070
                ;
000155'006005$         JSR     @GT.CH                   ;GET DELIMITER
000156'030434          LDA     2,QRCR
000157'146405          SUB     2,1,SNR
000160'000407          JMP     DT026
000161'006002$         JSR     @DIG.                    ;DECODE SECONDS
000162'044005-         STA     1,DTSC
000163'151005          MOV     2,2,SNR
000164'000434          JMP     DT070
000165'006007$         JSR     @CK.CR
000166'000743          JMP     DT020
                ;SET TIME
000167'020005-DT026:   LDA     0,DTSC
000170'024004-         LDA     1,DTMI
000171'030003-         LDA     2,DTHR
000172'006017          .SYSTEM
000173'021004          .STOD
000174'000426          JMP     DT080
000175'000403          JMP     DT040
                ;IF CHANGED, PRINT CURRENT
000176'020413 DT030:   LDA     0,DTCNG
000177'101004          MOV     0,0,SZR
000200'004427 DT040:   JSR     DT100
                ;RETURN
000201'020404          LDA     0,DT0A0
000202'024404          LDA     1,DT0A1
000203'030404          LDA     2,DT0A2
000204'006404          JSR     @DT0A3
                ;
000205'000000 DT0A0:   .-.
000206'000000 DT0A1:   .-.
000207'000000 DT0A2:   .-.
000210'000000 DT0A3:   .-.
                ;
```

```
0004 DT000
000211'000000 DTCNG: .-.      ;FLAG INDICATES CHANGE
000212'000015 QRCR:  15
                     ;
                     ;    ERROR ROUTINES
                     ;
000213'020412 DT050: LDA      0,Q05B
000214'000402        JMP      .+2
000215'020411 DT060: LDA      0,Q06B
000216'006003$       JSR      @P.CR
000217'000650        JMP      DT010
                     ;
000220'020405 DT070: LDA      0,Q05B
000221'000402        JMP      .+2
000222'020404 DT080: LDA      0,Q06B
000223'006003$       JSR      @P.CR
000224'000705        JMP      DT020
                     ;
000225'177777 Q05B:  Q13.2
000226'000074"Q06B:  Q06*2
                     .TITL    DT100
                     .ENT     DT100,PDAY,PTIM,PDATM
                     ;
                     ;    TYPE CURRENT DATE & TIME
                     ;
000227'054405 DT100: STA      3,DT1A3
000230'020403        LDA      0,QLCR
000231'004417        JSR      PDATM
000232'002402        JMP      @DT1A3
                     ;
000233'006400 QLCR:  6400
000234'000000 DT1A3: .-.
                     .TITL    DT200
                     .ZREL
                     .ENT     P.DA,P.TM,P.DT,G.DT,G.TM
000006-000235'P.DA:  PDAY
000007-000242'P.TM:  PTIM
000010-000250'P.DT:  PDATM
000011-000264'G.DT:  GTDT
000012-000325'G.TM:  GTTM
                     .NREL
                     ;GET,CONVERT & PRINT DATE
                     .ENT     PDAY
000235'040541 PDAY:  STA      0,Q04+6
000236'044423        STA      1,DT2A1
000237'054423        STA      3,DT2A3
000240'004424        JSR      GTDT
000241'000414        JMP      DT210
                     ;GET,CONVERT & PRINT TIME
                     .ENT     PTIM
000242'040542 PTIM:  STA      0,Q04+14
000243'044416        STA      1,DT2A1
000244'054416        STA      3,DT2A3
000245'004460        JSR      GTTM
000246'020415        LDA      0,Q04W6
000247'000407        JMP      DT220
                     ;GET,CONVERT & PRINT DATE & TIME
                     .ENT     PDATM
000250'040534 PDATM: STA      0,Q04+14
000251'044410        STA      1,DT2A1
000252'054410        STA      3,DT2A3
```

```
 0005 DT000
 01 000253'004411          JSR     GTDT
 02 000254'004451          JSR     GTTM
 03 000255'020506 DT210:   LDA     0,Q04B
 04 000256'006003$DT220:   JSR     @P.CR
 05 000257'024402          LDA     1,DT2A1
 06 000260'002402          JMP     @DT2A3
 07                  ;
 08 000261'000000 DT2A1:   .-.
 09 000262'000000 DT2A3:   .-.
 10 000263'000774"Q04W6:   Q04*2+14
                           .TITL   GTDT
 12                ;GET &  CONVERT DATE
 13                        .ENT    GTDT
 14 000264'040435 GTDT:    STA     0,DT3A0
 15 000265'044435          STA     1,DT3A1
 16 000266'050435          STA     2,DT3A2
 17 000267'054435          STA     3,DT3A3
 18                ;    GET DATE
 19 000270'006017          .SYSTEM
 20 000271'021006          .GDAY
 21 000272'000401          JMP     .+1
 22 000273'050000-         STA     2,DTYR
 23 000274'044001-         STA     1,DTMO
 24 000275'040002-         STA     0,DTDA
 25                ;
 26 000276'020464          LDA     0,Q03B
 27 000277'024000-         LDA     1,DTYR
 28 000300'006006$         JSR     @BN.AS
 29 000301'020465          LDA     0,Q03+2
 30 000302'040466          STA     0,Q04+0
 31                ;
 32 000303'020457          LDA     0,Q03B
 33 000304'024001-         LDA     1,DTMO
 34 000305'006006$         JSR     @BN.AS
 35 000306'020460          LDA     0,Q03+2
 36 000307'040463          STA     0,Q04+2
 37                ;
 38 000310'020452          LDA     0,Q03B
 39 000311'024002-         LDA     1,DTDA
 40 000312'006006$         JSR     @BN.AS
 41 000313'020453          LDA     0,Q03+2
 42 000314'040460          STA     0,Q04+4
 43                ;
 44              _ ;RETURN
 45 000315'020404          LDA     0,DT3A0
 46 000316'024404          LDA     1,DT3A1
 47 000317'030404          LDA     2,DT3A2
 48 000320'006404          JSR     @DT3A3
 49                ;
 50 000321'000000 DT3A0:   .-.
 51 000322'000000 DT3A1:   .-.
 52 000323'000000 DT3A2:   .-.
 53 000324'000000 DT3A3:   .-.
                           .TITL   GTTM
 55                ;GET &  CONVERT TIME
 56                        .ENT    GTTM
 57 000325'040774 GTTM:    STA     0,DT3A0
 58 000326'044774          STA     1,DT3A1
 59 000327'050774          STA     2,DT3A2
 60 000330'054774          STA     3,DT3A3
```

```
0006 DT000
              ;   GET TIME
000331'006017        .SYSTEM
000332'021003        .GTOD
000333'000401        JMP     .+1
000334'050003-       STA     2,DTHR
000335'044004-       STA     1,DTMI
000336'040005-       STA     0,DTSC
              ;
000337'020423        LDA     0,Q03B
000340'024003-       LDA     1,DTHR
000341'006006$       JSR     @BN.AS
000342'020424        LDA     0,Q03+2
000343'040433        STA     0,Q04+6
              ;
000344'020416        LDA     0,Q03B
000345'024004-       LDA     1,DTMI
000346'006006$       JSR     @BN.AS
000347'020417        LDA     0,Q03+2
000350'040430        STA     0,Q04+10
              ;
000351'020411        LDA     0,Q03B
000352'024005-       LDA     1,DTSC
000353'006006$       JSR     @BN.AS
000354'020412        LDA     0,Q03+2
000355'040425        STA     0,Q04+12
              ;
              ;RETURN
000356'020743        LDA     0,DT3A0
000357'024743        LDA     1,DT3A1
000360'030743        LDA     2,DT3A2
000361'006743        JSR     @DT3A3
              ;STORAGE
                     .ENT    DT002
              DT002:
000362'000750"Q03B:  Q03*2
000363'000760"Q04B:  Q04*2
000364'054130 Q03:   .TXT    /XXXXXX/
       054130
       054130
       000000
000370'054130 Q04:   .TXT    /XXY XXM XXD XXH XXM XXS   <15>/
       054440
       054130
       046440
       054130
       042040
       054130
       044040
       054130
       046440
       054130
       051440
       020040
       006400
                     .TITL   DT000
                     .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
 0007 DT000

SN.AS 000006$ XD    1/03    5/28    5/34    5/40    6/11    6/17    6/23
CK.CR 000007$ XD    1/04    2/53    3/36
DIG.  000002$ XD    1/03    2/30    2/39    2/48    3/14    3/23    3/32
DT000 000051' EN    1/01    2/01    2/02    2/06    6/55
DT001 000000' EN    1/17    1/18
DT002 000362' EN    6/33    6/34
DT010 000067'       2/24    2/54    4/10
DT020 000131' EN    2/27    3/04    3/05    3/37    4/16
DT026 000167'       3/22    3/31    3/39
DT030 000176'       3/08    3/47
DT040 000200'       3/45    3/49
DT050 000213'       2/33    2/38    2/42    2/47    2/51    4/06
DT060 000215'       3/01    4/08
DT070 000220'       3/17    3/26    3/35    4/12
DT080 000222'       3/44    4/14
DT0A0 000205'       2/06    3/51    3/56
DT0A1 000206'       2/07    3/52    3/57
DT0A2 000207'       2/08    3/53    3/58
DT0A3 000210'       2/09    3/54    3/59
DT100 000227' EN    2/22    3/49    4/20    4/21    4/25
DT1A3 000234'       4/25    4/28    4/31
DT210 000255'       4/47    5/03
DT220 000256'       4/55    5/04
DT2A1 000261'       4/44    4/51    4/59    5/05    5/08
DT2A3 000262'       4/45    4/52    4/60    5/06    5/09
DT3A0 000321'       5/14    5/45    5/50    5/57    6/28
DT3A1 000322'       5/15    5/46    5/51    5/58    6/29
DT3A2 000323'       5/16    5/47    5/52    5/59    6/30
DT3A3 000324'       5/17    5/48    5/53    5/60    6/31
DTCNG 000211'       2/15    3/02    3/47    4/01
DTDA  000002- EN    1/07    1/10    2/49    2/56    5/24    5/39
DTHR  000003- EN    1/07    1/11    3/15    3/41    6/05    6/10
DTMI  000004- EN    1/07    1/12    3/12    3/24    3/40    6/06    6/16
DTMO  000001- EN    1/07    1/09    2/40    2/57    5/23    5/33
DTSC  000005- EN    1/07    1/13    3/11    3/33    3/39    6/07    6/22
DTYR  000000- EN    1/07    1/08    2/31    2/58    5/22    5/27
GTDT  000264' EN    4/38    4/46    5/01    5/11    5/13    5/14
GTTM  000325' EN    4/39    4/53    5/02    5/54    5/56    5/57
GT.CH 000005$ XD    1/03    2/35    2/44    3/19    3/28
G.DT  000011- EN    4/34    4/38
G.TM  000012- EN    4/34    4/39
IBUF  000001$ XD    1/03    2/29    3/13
PDATM 000250' EN    4/21    4/27    4/37    4/57    4/58
PDAY  000235' EN    4/21    4/35    4/42    4/43
PTIM  000242' EN    4/21    4/36    4/49    4/50
P.CR  000003$ XD    1/03    2/12    2/25    3/06    4/09    4/15    5/04
P.DA  000006- EN    4/34    4/35
P.DT  000010- EN    4/34    4/37
P.TM  000007- EN    4/34    4/36
Q00   000000'       1/19    1/57
Q00B  000045'       1/57    2/11
Q01   000015'       1/33    1/58
Q01B  000046'       1/58    2/24
Q02   000026'       1/41    1/59
Q02B  000047'       1/59    3/05
Q03   000364'       5/29    5/35    5/41    6/12    6/18    6/24    6/35
                    6/37
Q03B  000362'       5/26    5/32    5/38    6/09    6/15    6/21    6/35
Q04   000370'       4/43    4/50    4/58    5/10    5/30    5/36    5/42
```

```
 0008 DT000

                          6/13    6/19    6/25    6/36    6/41
Q04B  000363'             5/03    6/36
Q04W6 000263'             4/54    5/10
Q05B  000225'             4/06    4/12    4/18
Q06   000036'             1/49    4/19
Q06B  000226'             4/08    4/14    4/19
Q13.2 000225' XN          1/02    4/18
QLCR  000233'             4/26    4/30
QRCR  000212'             2/36    2/45    3/20    3/29    4/02
R.LN  000004$ XD          1/03    2/26    3/07
VB100 000050' XN          1/02    1/60
VBSUB 000050'             1/60    2/13
```

```
0001 SP000 MACRO REV 06.50             19:23:42 10/31/80
                        .TITL   SP000   ;STORAGE
                        .ENT    Q01.2,Q07.2,Q08.2
                  ;
                        .EXTD   BN.AS,GT.CH,R.LN,P.CR,R.PS,DIV.I,FM.ER
                        .EXTD   IBUF,MVINI,WRN.A,G.ADR,CK.CR
                        .EXTD   TAB1A,TAB1I,ADR,HIADR,MNADR,MNBAD
                        .EXTD   RAT1A,RAT2A,RAT3A,RAT4A,RAT5A,RAT6A,RAT7A
                        .EXTD   ERCT,IFMOD,BFPT,LADR1,LADR2,LINCR,TNUM0
                  ;
                        .EXTN   RATE0,RATE1,RATE2,RATE3,RATE4,RATE5,RATE6
                        .EXTN   RATE7,RATEM,VB200,VB220,VB240
                  ;
       000001           .TXTM   1
                  ;
                        .ZREL
                        .ENT    TNUM
000000-000000 TNUM:     .-.     ;SAMPLES/SEC USED
                        .NREL
                        .ENT    SP001
              SP001:
000000'005040 Q00:      .TXT    /<12> * * * SAMPLING PATTERN * * *<15>/
       025040
       025040
       025040
       051501
       046520
       046111
       047107
       020120
       040524
       052105
       051116
       020052
       020052
       020052
       006400
000020'044516 Q01:      .TXT    /INITIALIZE, MODIFY, OR CHECK: /
       044524
       044501
       046111
       055105
       026040
       046517
       042111
       043131
       026040
       047522
       020103
       044105
       041513
       035040
       000000
000040'041517 Q02:      .TXT    /CONTINUING DESTROYS THE SAMPLING PATTERN, /
       047124
       044516
       052511
       047107
       020104
       042523
       052122
```

```
 0002 SP000
01         047531
02         051440
03         052110
04         042440
05         051501
06         046520
07         046111
08         047107
09         020120
10         040524
11         052105
12         051116
13         026040
14         000000
15 000066'046517 Q03:    .TXT    /MONITORING DELETED.<15>/
16         047111
17         052117
18         051111
19         047107
20         020104
21         042514
22         042524
23         042504
24         027015
25         000000
26 000101'040514 Q04:    .TXT    /ALARM B DELETED.<15>/
27         040522
28         046440
29         041040
30         042105
31         046105
32         052105
33         042056
34         006400
35 000112'042516 Q06:    .TXT    /ENTER ADRS OR ADRS,RATE,MODE:<15>/
36         052105
37         051040
38         040504
39         051123
40         020117
41         051040
42         040504
43         051123
44         026122
45         040524
46         042454
47         046517
48         042105
49         035015
50         000000
51 000132'042516 Q07:    .TXT    /ENTER ADRS:<15>/
52         052105
53         051040
54         040504
55         051123
56         035015
57         000000
58 000141'035040 Q08:    .TXT    /: /
59         000000
60 000143'051101 Q14:    .TXT    /RATE INVALID<15>/
```

```
0003 SP000
        052105
        020111
        047126
        040514
        044504
        006400
000152'046517 Q15:    .TXT    /MODE INVALID<15>/
        042105
        020111
        047126
        040514
        044504
        006400
000161'052117 Q16:    .TXT    *TOO MANY SAMPLES/SECOND<15>*
        047440
        046501
        047131
        020123
        040515
        050114
        042523
        027523
        042503
        047516
        042015
        000000

000176'054130 Q21:    .TXT    *XXXXXX SAMPLES/SECOND TO GO<15>*
        054130
        054130
        020123
        040515
        050114
        042523
        027523
        042503
        047516
        042040
        052117
        020107
        047415
        000000
000215'005123 Q22:    .TXT    /<12>SAMPLING PATTERN 'A' LOADED<15>/
        040515
        050114
        044516
        043440
        050101
        052124
        042522
        047040
        023501
        023440
        046117
        040504
        042504
        006400
        000040"Q01.2= Q01*2
        000264"Q07.2= Q07*2
        000302"Q08.2= Q08*2
```

```
                                 .TITL   SP000   ;PRELIMINARY
                                 .ENT    SP000
                        ; * * * SAMPLING PATTERN SELECTION * * *
                        ;
000234'040475 SP000:    STA      0,SP0A0
000235'044475           STA      1,SP0A1
000236'050475           STA      2,SP0A2
000237'054475           STA      3,SP0A3
                        ;
000240'020513           LDA      0,Q00B
000241'006004$          JSR      @P.CR
                        ;                              ;TYPE "SAMPLING PATTERN"
000242'020011$          LDA      0,MVINI
000243'101005           MOV      0,0,SNR               ;CHECK MACRO MODE
000244'000421           JMP      SP015
                        ;ASK "INIT, MOD, OR CK?"       ;CHECK ONLY
000245'006470           JSR      @VB1SB
000246'020506 SP010:    LDA      0,Q01B                ;EXPLAIN INIT,MOD, OR CK
000247'006004$          JSR      @P.CR
                        ;                              ;TYPE "INIT,MOD OR CK?"
000250'006003$          JSR      @R.LN                 ;READ A LINE
000251'002457           JMP      @SP093                ;JUST CR, GO TO NEXT SEC
000252'020010$          LDA      0,IBUF
000253'006002$          JSR      @GT.CH
                        ;                              ;GET CHARACTER
000254'020464           LDA      0,QRI
000255'122405           SUB      1,0,SNR
000256'000413           JMP      SP020
                        ;                              ;INITIALIZE
000257'020462           LDA      0,QRM
000260'122405           SUB      1,0,SNR
000261'000435           JMP      SP080
                        ;                              ;MODIFY
000262'020460           LDA      0,QRC
000263'122404           SUB      1,0,SZR
000264'000762           JMP      SP010
                        ;CHECK ONLY - DON'T HALT PROCESSING  ;TRY AGAIN
000265'040033$SP015:    STA      0,IFMOD               ;ZERO FLAG FOR NO MODS
000266'006451           JSR      @VB3SB                ;EXPLAIN ADR
000267'020470           LDA      0,Q07B                ;TYPE "ENTER ADR"
000270'000432           JMP      SP090
                        ;INITIALIZE
000271'020464 SP020:    LDA      0,Q02B                ;WARNING MESSAGE
000272'006012$          JSR      @WRN.A                ;CHANCE TO ABORT
000273'000753           JMP      SP010                 ;ABORT
                        ; LOAD SAMPLING PATTERN 'A'
000274'020467           LDA      0,Q22B
000275'006004$          JSR      @P.CR
                        ; DEFINE RATE/MODE
       002000 P1=       1B5        ;RATE=1,MODE=P
       004000 P2=       2B5
       006000 P3=       3B5
       010000 P4=       4B5
       012000 P5=       5B5
       014000 P6=       6B5
       016000 P7=       7B5
                        ;
       003000 R1=       1B5+1B6
       005000 R2=       2B5+1B6
       007000 R3=       3B5+1B6
```

```
0005 SP000
        011000 R4=     4B5+1B6
        013000 R5=     5B5+1B6
        015000 R6=     6B5+1B6
        017000 R7=     7B5+1B6
        000000 Z0=     0
                  ;
                  ;
                  ;   START INITIALIZATION
   000276'034427          LDA     3,TABAA ;TABA ADR
   000277'032427          LDA     2,@TABQ
   000300'024020$         LDA     1,HIADR
   000301'044037$         STA     1,LINCR
   000302'024450          LDA     1,TNUMA ;INIT # SAMPLES/SEC
   000303'044000-         STA     1,TNUM
   000304'102400          SUB     0,0      ;CHANNEL NO.
                  ;
                          .TITL   SP030
                          .ENT    SP030
                  ;   LOOP TO FILL TABLE TAB1
                  SP030:
   000305'025400          LDA     1,0,3   ;RATE/MODE INFO
   000306'107000          ADD     0,1             ;COMBINE INFO
   000307'045000          STA     1,0,2   ;STORE INFO >TAB1
   000310'101400          INC     0,0             ;NEXT CHANNEL
   000311'151400          INC     2,2     ;NEXT TAB1 LOCN.
   000312'175400          INC     3,3     ;NEXT TABA LOCN.
   000313'014037$         DSZ     LINCR           ;LIST FOR COMPLETION
   000314'000771          JMP     SP030
   000315'000401          JMP     SP080
                  ;
                  ;    REPETITION MACRO
                          .MACRO  REPT
                          .DO     ^1
                          ^2
                          .ENDC
                  %
                  ;
                  ;  INIT OR MODIFY
                          .ENT    SP080
   000316'102520 SP080:  SUBZL   0,0                  ;CONSTANT 1
   000317'040033$         STA     0,IFMOD              ;SET FLAG TO PERMIT MODS
                  ;
   000320'006416          JSR     @VB2SB               ;EXPLAIN SP SELECTION
   000321'020435          LDA     0,006B               ;TYPE "ENTER ADR,RATE,MODE"
   000322'006004$SP090:  JSR     @P.CR
   000323'006404          JSR     @SP091               ;TYPE # TO GO
   000324'000440          JMP     SP100
                  ;
   000325'000727'TABAA:  TABA
   000326'000015$TABQ:   TAB1A
   000327'000600'SP091:  SP450
   000330'000631'SP093:  SP800
   000331'000000 SP0A0:  .-.
   000332'000000 SP0A1:  .-.
   000333'000000 SP0A2:  .-.
   000334'000000 SP0A3:  .-.
   000335'177777 VB1SB:  VB200
   000336'177777 VB2SB:  VB220
   000337'177777 VB3SB:  VB240
                          .TITL   SP002   ;STORAGE
```

```
 0006 SP000
01                                 .ENT    SP002
02                         ;CONSTANTS & STORAGE
03                         SP002:
04 000340'000111 QRI:    "I
05 000341'000115 QRM:    "M
06 000342'000103 QRC:    "C
07 000343'000122 QRR:    "R
08 000344'000120 QRP:    "P
09 000345'000054 QRCM:   ",
10 000346'000060 QR0:    "0
11 000347'000067 QR7:    "7
12 000350'000000 RATE:   .-.
13 000351'000000 MODE:   .-.
14 000352'026145 TNUMA:  11365.   ;SAMPLES/SEC FOR PATTERN 'A'
15 000353'000000"Q00B:   Q00*2
16 000354'000040"Q01B:   Q01*2
17 000355'000100"Q02B:   Q02*2
18 000356'000224'Q06B:   Q06*2
19 000357'000264"Q07B:   Q07*2
20 000360'000302"Q08B:   Q08*2
21 000361'000306"Q14B:   Q14*2
22 000362'000324"Q15B:   Q15*2
23 000363'000432"Q22B:   Q22*2
24                       ;
25                       ;

                                .TITL   SP100   ;PROCESS
27                              .ENT    SP100
28                       ;ROUTINE TO LOOP FOR SAMPLING PATTERN MODIFICATION
29 000364'020774 SP100: LDA     0,Q08B
30 000365'006004$        JSR     @P.CR               ;ISSUE PROMPT
31                       ;READ LINE - INITIALIZE
32 000366'006003$        JSR     @R.LN
33 000367'002741         JMP     @SP093
34                       ;GET& CHECK ADR(S)
35 000370'006013$        JSR     @G.ADR
36 000371'000773         JMP     SP100
37                       ;GET & CHECK RATE
38 000372'006002$        JSR     @GT.CH
39 000373'034753         LDA     3,QR0
40 000374'030753         LDA     2,QR7
41 000375'146033         ADCZ#   2,1,SNC
42 000376'136032         ADCZ#   1,3,SZC
43 000377'000402         JMP     .+2       -         ;ERROR
44 000400'000406         JMP     SP120               ;OK
45                       ;
46 000401'040034$        STA     0,BFPT
47 000402'020757         LDA     0,Q14B
48 000403'006004$        JSR     @P.CR               ;TYPE "RATE INVALID"
49 000404'010032$        ISZ     ERCT
50 000405'020034$        LDA     0,BFPT
51                       ;
52 000406'166400 SP120: SUB     3,1
53 000407'044741         STA     1,RATE
54 000410'125004         MOV     1,1,SZR
55 000411'000404    /    JMP     SP130
56                 ;       IF ZERO RATE, MAY BE CR
57 000412'006014$        JSR     @CK.CR              ;GET AND CHECK FOR CR
58 000413'000403         JMP     SP140               ;ALREADY GOT CHAR
59 000414'000424         JMP     SP160               ;END OF LINE
60                       ;GET & CHECK COMMA
```

```
000415'006002$SP130: JSR      @GT.CH
000416'030727 SP140: LDA      2,QRCM
000417'132404        SUB      1,2,SZR
000420'006007$       JSR      @FM.ER          ;TYPE "FORMAT ERROR"
                ;GET & CHECK MODE
000421'006002$       JSR      @GT.CH
000422'030721        LDA      2,QRR
000423'132405        SUB      1,2,SNR         ;CHECK FOR "R"
000424'000411        JMP      SP150
                ;
000425'030717        LDA      2,QRP
000426'132405        SUB      1,2,SNR         ;CHECK FOR "P"
000427'000406        JMP      SP150
                ;
000430'040034$       STA      0,BFPT
000431'020731        LDA      0,Q15B          ;TYPE "MODE INVALID"
000432'006004$       JSR      @P.CR
000433'010032$       ISZ      ERCT
000434'020034$       LDA      0,BFPT
                ;
000435'044714 SP150: STA      1,MODE          ;SAVE R OR P
                ;GET & CHECK CR
000436'006014$       JSR      @CK.CR
000437'000401        JMP      .+1
                ;CHECK FOR ANY ERRORS IN LINE
                     .ENT     SP160
000440'024032$SP160: LDA      1,ERCT
000441'125004        MOV      1,1,SZR
000442'000722        JMP      SP100
                ;IF ZERO RATE, CHECK MONITORING & ALARM "B"
000443'024035$       LDA      1,LADR1
000444'030036$       LDA      2,LADR2
000445'034703        LDA      3,RATE
000446'175004        MOV      3,3,SZR
000447'000421        JMP      SP170
                ;    CHECK MONITORING
000450'034021$       LDA      3,MNADR
000451'156033        ADCZ#    2,3,SNC
000452'166032        ADCZ#    3,1,SZC
000453'000405        JMP      SP164           ;MONITORING OK
                ;    DELETE MONITORING
000454'176400        SUB      3,3
000455'054021$       STA      3,MNADR
000456'020455        LDA      0,Q03B
000457'006004$       JSR      @P.CR
                ;    CHECK ALARM B
000460'034022$SP164: LDA      3,MNBAD
000461'156033        ADCZ#    2,3,SNC
000462'166032        ADCZ#    3,1,SZC
000463'000405        JMP      SP170           ;ALARM B OK
                ;    DELETE ALARM B
000464'176400        SUB      3,3
000465'054022$       STA      3,MNBAD
000466'020446        LDA      0,Q04B
000467'006004$       JSR      @P.CR
                ;LOOP TO MODIFY TABLE
                     .ENT     SP170
000470'030036$SP170: LDA      2,LADR2
000471'132433        SUBZ#    1,2,SNC
000472'000437        JMP      SP190
```

```
0008 SP000
                ;SET UP REQUIRED STORAGE
000473'044017$          STA     1,ADR
000474'034015$          LDA     3,TAB1A
000475'137000           ADD     1,3
000476'054016$          STA     3,TAB1I
                ;CHECK UPDATED SAMPLES PER SEC
000477'020651           LDA     0,RATE
000500'026016$          LDA     1,@TAB1I          ;GET OLD RATE
000501'030435           LDA     2,MKRAT
000502'147700           ANDS    2,1
000503'125220           MOVZR   1,1
000504'125220           MOVZR   1,1
000505'004432           JSR     SP400             ;COMPUTE # TO GO
000506'000422           JMP     SP180             ;ERROR RETURN
                ;MODIFY TABLE FOR THIS ADR
                ;    ACCUMULATE WORD IN AC0
000507'024642           LDA     1,MODE            ;GET MODE
000510'020634           LDA     0,QRP             ;CHECK WHICH
000511'122404           SUB     1,0,SZR           ;USE ZERO (PEAK)
000512'020423           LDA     0,MKMOD           ;GET BIT (REAL)
                ;
000513'024635           LDA     1,RATE            ;GET RATE
000514'125300           MOVS    1,1               ;MOVE TO BITS 3-5
000515'125120           MOVZL   1,1
000516'125120           MOVZL   1,1
000517'123000           ADD     1,0               ;ADD TO WORD
                ;
000520'034017$          LDA     3,ADR             ;GET ADR
000521'163000           ADD     3,0
000522'042016$          STA     0,@TAB1I          ;SAVE REVISED WORD
                ;INCREMENT LOOP
000523'024035$          LDA     1,LADR1
000524'020037$          LDA     0,LINCR
000525'107000           ADD     0,1
000526'044035$          STA     1,LADR1
000527'000741           JMP     SP170
                ;FOR ERROR, REPORT UNCHANGED STATUS
000530'006005$SP180:    JSR     @R.PS
                ;TYPE SAMPLES TO GO
000531'004447 SP190:    JSR     SP450
000532'000632           JMP     SP100
                ;
000533'000154"Q03B:     Q03*2
000534'000202"Q04B:     Q04*2
000535'001000 MKMOD:    1B6
000536'016000 MKRAT:    7B5
                        .TITL   SP400    ;MAX SAMPLING
                        .ENT    SP400
                ;ROUTINE TO COMPUTE # TO GO
                ;    AC0  NEW RATE
                ;    AC1  OLD RATE
000537'040435 SP400:    STA     0,SP4A0           ;SAVE ACCUMULATORS
000540'044435           STA     1,SP4A1
000541'050435           STA     2,SP4A2
000542'054435           STA     3,SP4A3
                ;
000543'030000-          LDA     2,THUM            ;GET CURRENT SUM
000544'106415           SUB#    0,1,SNR           ;CHECK FOR NO CHANGE
000545'000411           JMP     SP410
                ;
```

```
 0009 SP000
000546'034462          LDA     3,ADRT0
000547'117000          ADD     0,3
000550'021400          LDA     0,0,3
000551'113000          ADD     0,2             ;ADD NEW SAMPLES/SEC
               ;
000552'034456          LDA     3,ADRT0
000553'137000          ADD     1,3
000554'021400          LDA     0,0,3
000555'112400          SUB     0,2             ;SUB OLD SAMPLES/SEC
               ;CHECK STILL VALID
000556'024441 SP410:   LDA     1,TMAXM
000557'146423          SUBZ    2,1,SNC         ;SKIP IF AC2<=AC1
000560'000403          JMP     SP420           ;AC2>AC1
000561'050000-         STA     2,TNUM          ;SAVE NEW SUM
000562'000405          JMP     SP430
               ;ERROR
000563'014414 SP420:   DSZ     SP4A3           ;RETURN
000564'020427          LDA     0,Q16B
000565'006004$         JSR     @P.CR
000566'010032$         ISZ     ERCT
               ;RETURN TO CALLER
000567'020405 SP430:   LDA     0,SP4A0
000570'024405          LDA     1,SP4A1
000571'030405          LDA     2,SP4A2
000572'034405          LDA     3,SP4A3
000573'001401          JMP     1,3             ;RETURN+1
               ;
000574'000000 SP4A0:   .-.
000575'000000 SP4A1:   .-.
000576'000000 SP4A2:   .-.
000577'000000 SP4A3:   .-.
                       .TITL   SP450   ;SAMPLES TO GO
                       .ENT    SP450
               ;PRINT SAMPLES TO GO
000600'054777 SP450:   STA     3,SP4A3
               ;
000601'024416          LDA     1,TMAXM
000602'030000-         LDA     2,TNUM
000603'146400          SUB     2,1
000604'030412          LDA     2,D10
000605'006006$         JSR     @DIV.I          ;DIVIDE BY 10
000606'020406          LDA     0,Q21B
000607'006001$         JSR     @BN.AS          ;CONVERT TO ASCII
000610'020405          LDA     0,Q21B2
000611'006004$         JSR     @P.CR           ;TYPE "# TO GO"
000612'002765          JMP     @SP4A3
               ;
000613'000342"Q16B:    Q16*2
000614'000374"Q21B:    Q21*2
000615'000376"Q21B2:   Q21*2+2
000616'000012 D10:     10.
                       .ENT    TMAXM
000617'177777 TMAXM:   RATEM
000620'177777 T0:      RATE0
000621'177777          RATE1
000622'177777          RATE2
000623'177777          RATE3
000624'177777          RATE4
000625'177777          RATE5
000626'177777          RATE6
```

```
0010 SP000
000627'177777          RATE7
000630'000620'ADRT0:   T0
                       .TITL   SP800    ;SORT
                       .ENT    SP800
                   ;
                   ;   SORT ADR TABLE BY TIME OPTION
                   ;
000631'020023$SP800:   LDA      0,RAT1A           ;MOVE TABLE ADRESSES
000632'040456          STA      0,TMT1+1          ;TO INDIR TABLE
000633'020024$         LDA      0,RAT2A
000634'040455          STA      0,TMT1+2
000635'020025$         LDA      0,RAT3A
000636'040454          STA      0,TMT1+3
000637'020026$         LDA      0,RAT4A
000640'040453          STA      0,TMT1+4
000641'020027$         LDA      0,RAT5A
000642'040452          STA      0,TMT1+5
000643'020030$         LDA      0,RAT6A
000644'040451          STA      0,TMT1+6
000645'020031$         LDA      0,RAT7A
000646'040450          STA      0,TMT1+7
                   ;SET UP FOR MAIN LOOP (SORT)
000647'030015$         LDA      2,TAB1A           ;GET FETCH ADR FOR SORT
000650'020020$         LDA      0,HIADR           ;LOOP COUNT
000651'040446          STA      0,SP891
000652'151400          INC      2,2               ;SKIP ZERO ADR
000653'014444          DSZ      SP891
                   ;   LOOP TO SORT
000654'021000 SP820:   LDA      0,0,2             ;GET NEXT WORD
000655'024661          LDA      1,MKRAT           ;MASK FOR RATE
000656'107705          ANDS     0,1,SNR           ;ISOLATE & SHIFT RATE
000657'000411          JMP      SP830             ;SKIP IF RATE ZERO
000660'125220          MOVZR    1,1               ;RJ RATE
000661'125220          MOVZR    1,1               ;FOR INDEXING
000662'034424          LDA      3,TMT1A           ;GET ADR START INDIR TABLE
000663'137000          ADD      1,3               ;INDEX
000664'024421          LDA      1,MKTAB           ;MASK FOR MODE/ADR
000665'123400          AND      1,0
000666'043400          STA      0,@0,3            ;SAVE IN PROPER TABLE
000667'011400          ISZ      0,3               ;INCR STORE ADDRESS
                   ;    LOOP INCR
000670'151400 SP830:   INC      2,2               ;INCR FETCH ADR
000671'014426          DSZ      SP891             ;COUNT LOOP
000672'000762          JMP      SP820
                   ;    ZERO NEXT WORD OF EACH TABLE
000673'102400          SUB      0,0               ;GET ZERO
000674'034412          LDA      3,TMT1A           ;GET TABLE ADR
000675'043401          STA      0,@1,3
000676'043402          STA      0,@2,3
000677'043403          STA      0,@3,3
000700'043404          STA      0,@4,3
000701'043405          STA      0,@5,3
000702'043406          STA      0,@6,3
000703'043407          STA      0,@7,3
                   ;
000704'000414          JMP      SP900             ;EXIT THIS ROUTINE
                   ;STORAGE
000705'001777 MKTAB:   1777               ;KEEP ONLY MODE/ADR BITS
000706'000707'TMT1A:   TMT1
000707'000010 TMT1:    .BLK     10        ;TABLE FOR INDIRECT ADDRESSES
```

```
0011 SP000
000717'000000 SP891: .-.                   ;LOOP COUNTER
                     ;
                     ;    END OF SAMPLING PATTERN SELECTION
                     ;
000720'034406 SP900: LDA     3,SP990
000721'021400        LDA     0,0,3
000722'025401        LDA     1,1,3
000723'031402        LDA     2,2,3
000724'035403        LDA     3,3,3
000725'001400        JMP     0,3
                     ;
000726'000331'SP990: SP0A0
                     ;
       000001        .NOMAC  1
                     .TITL   TABA
                     .ENT    TABA
000727'004000 TABA:  2B5                       ;ADR 0
                     REPT    2       P2        ;1-2
                     REPT    2       Z0        ;3-4
000734'003000        R1                        ;5
                     REPT    5       Z0        ;5-10
                     REPT    3       R3        ;11-13
                     REPT    3       Z0        ;14-16
000750'004000        P2                        ;17
000751'000000        Z0                        ;18
                     REPT    3       P2        ;19-21
                     REPT    4       Z0        ;22-25
000761'004000        P2                        ;26
                     REPT    2       Z0        ;27-28
000764'004000        P2                        ;29
000765'000000        Z0                        ;30
000766'004000        P2                        ;31
                     REPT    12.     P2        ;32-43
001003'000000        Z0                        ;44
                     REPT    2       P2        ;45-46
001006'000000        Z0                        ;47
                     REPT    4       P2        ;48-51
                     REPT    2       R5        ;52-53
                     REPT    7       P2        ;54-60
                     REPT    2       Z0        ;61-62
001026'004000        P2                        ;63
                     REPT    2       R5        ;64-65
                     REPT    11.     P2        ;66-76
001044'000000        Z0                        ;77
001045'004000        P2                        ;78
001046'000000        Z0                        ;79
                     REPT    4       P2        ;80-83
001053'000000        Z0                        ;84
                     REPT    2       P2        ;85-86
                     REPT    2       Z0        ;87-88
                     REPT    3       P2        ;89-91
                     REPT    3       Z0        ;92-94
001066'004000        P2                        ;95
001067'000000        Z0                        ;96
                     REPT    2       P2        ;97-98
                     REPT    2       Z0        ;99-100
                     REPT    6       P2        ;101-106
001102'000000        Z0                        ;107
001103'004000        P2                        ;108
001104'013000        R5                        ;109
```

```
0012 SP000
                            REPT    4       P2      ;110-113
                            REPT    2       Z0      ;114-115
                            REPT    3       P2      ;116-118
   001116'000000            Z0                      ;119
                            REPT    2       P2      ;120-121
                            REPT    2       Z0      ;122-123
                            REPT    2       P2      ;124-125
                            REPT    4       P2      ;126-129
   001131'000000            Z0                      ;130
                            REPT    4       P2      ;131-134
   001136'000000            Z0                      ;135
   001137'004000            P2                      ;136
   001140'000000            Z0                      ;137
   001141'004000            P2                      ;138
   001142'000000            Z0                      ;139
   001143'004000            P2                      ;140
                            REPT    4       Z0      ;141-144
   001150'004000            P2                      ;145
   001151'000000            Z0                      ;146
                            REPT    3       P2      ;147-149
                            REPT    2       Z0      ;150-151
                            REPT    2       P2      ;152-153
   001161'000000            Z0                      ;154
   001162'004000            P2                      ;155
   001163'000000            Z0                      ;156
                            REPT    2       P2      ;157-158
   001166'000000            Z0                      ;159
   001167'004000            P2                      ;160
   001170'000000            Z0                      ;161
                            REPT    2       P2      ;162-163
                            REPT    2       R5      ;164-165
   001175'004000            P2                      ;166
                            REPT    2       Z0      ;167-168
                            REPT    2       P2      ;169-170
                            REPT    2       Z0      ;171-172
   001204'013000            R5                      ;173
   001205'004000            P2                      ;174
   001206'000000            Z0                      ;175
                            REPT    4       P2      ;176-179
                            REPT    7       Z0      ;180-186
                            REPT    2       R1      ;187-188
                            REPT    9.      Z0      ;189-197
   001235'011000            R4                      ;198
   001236'003000            R1                      ;199
   001237'005000            R2                      ;200
   001240'011000            R4                      ;201
                            REPT    2       R2      ;202-203
                            REPT    3       R1      ;204-206
   001246'000000            Z0                      ;207
   001247'011000            R4                      ;208
   001250'006000            P3                      ;209
                            REPT    2       Z0      ;210-211
                            REPT    3       P1      ;212-214
                            REPT    3       R3      ;215-217
                            REPT    6       Z0      ;218-223
                            REPT    3       R1      ;224-226
   001272'000000            Z0                      ;227
                            REPT    7       R1      ;228-234
   001302'000000            Z0                      ;235
                            REPT    3       R1      ;236-238
```

```
 0013 SP000
01                           REPT    5        Z0        ;239-243
02                           REPT    2        R2        ;244-245
03                           REPT    6        Z0        ;246-251
04                           REPT    2        R1        ;252-253
05 001325'000000             Z0                         ;254
06 001326'005000             R2                         ;255
07                 ;
                             .TITL   SP000
09                           .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
  0014 SP000

ADR    000017$ XD    1/06    8/02    8/28
ADRT0  000630'       9/01    9/06   10/02
BFPT   000034$ XD    1/08    6/46    6/50    7/15    7/19
 N.AS  000001$ XD    1/04    9/43
CK.CR  000014$ XD    1/05    6/57    7/23
D10    000615'       9/40    9/51
DIV.I  000006$ XD    1/04    9/41
ERCT   000032$ XD    1/08    6/49    7/18    7/27    9/20
FM.ER  000007$ XD    1/04    7/04
GT.CH  000002$ XD    1/04    4/24    6/38    7/01    7/06
G.ADR  000013$ XD    1/05    6/35
HIADR  000020$ XD    1/06    5/11   10/24
IBUF   000010$ XD    1/05    4/23
IFMOD  000033$ XD    1/08    4/38    5/41
LADR1  000035$ XD    1/08    7/31    8/32    8/35
LADR2  000036$ XD    1/08    7/32    7/58
LINCR  000037$ XD    1/08    5/12    5/27    8/33
MKMOD  000535'       8/20    8/45
MKRAT  000536'       8/09    8/46   10/30
MKTAB  000705'      10/37   10/58
MNADR  000021$ XD    1/06    7/37    7/43
MNBAD  000022$ XD    1/06    7/47    7/53
MODE   000351'       6/13    7/21    8/17
MVINI  000011$ XD    1/05    4/13
P1     002000        4/50   12/54
P2     004000        4/51   11/19   11/24   11/27   11/28   11/30   11/32
                    11/34   11/36   11/38   11/40   11/41   11/44   11/45
                    11/48   11/50   11/52   11/53   11/56   11/58   11/59
                    12/02   12/04   12/06   12/08   12/09   12/11   12/12
                    12/14   12/16   12/18   12/21   12/23   12/24   12/27
                    12/28   12/31   12/32   12/35   12/37   12/40
P3     006000        4/52   12/51
P4     010000        4/53
P5     012000        4/54
P6     014000        4/55
P7     016000        4/56
P.CR   000004$ XD    1/04    4/11    4/19    4/48    5/45    6/30    6/48
                     7/17    7/45    7/55    9/19    9/45
Q00    000000'       1/21    6/15
Q00B   000353'       4/10    6/15
Q01    000020'       1/37    3/58    6/16
Q01B   000354'       4/18    6/16
Q01.2  000040" EN    1/02    3/58
Q02    000040'       1/53    6/17
Q02B   000355'       4/43    6/17
Q03    000066'       2/15    8/43
Q03B   000533'       7/44    8/43
Q04    000101'       2/26    8/44
Q04B   000534'       7/54    8/44
Q06    000112'       2/35    6/18
Q06B   000356'       5/44    6/18
Q07    000132'       2/51    3/59    6/19
Q07B   000357'       4/40    6/19
Q07.2  000264" EN    1/02    3/59
Q08    000141'       2/58    3/60    6/20
Q08B   000360'       6/20    6/29
Q08.2  000302" EN    1/02    3/60
Q14    000143'       2/60    6/21
Q14B   000361'       6/21    6/47
```

```
  0015 SP000

 Q15    000152'         3/07    6/22
 Q15B   000362'         6/22    7/16
 Q16    000161'         3/14    9/48
 Q16B   000613'         9/18    9/48
 Q21    000176'         3/28    9/49    9/50
 Q21B   000614'         9/42    9/49
 Q21B2  000615'         9/44    9/50
 Q22    000215'         3/43    6/23
 Q22B   000363'         4/47    6/23
 QR0    000346'         6/10    6/39
 QR7    000347'         6/11    6/40
 QRC    000342'         4/34    6/06
 QRCM   000345'         6/09    7/02
 QRI    000340'         4/26    6/04
 QRM    000341'         4/30    6/05
 QRP    000344'         6/08    7/11    8/18
 QRR    000343'         6/07    7/07
 R1     003000          4/58   11/20   12/42   12/44   12/49   12/57   12/59
                       13/01   13/05
 R2     005000          4/59   12/45   12/48   13/03   13/06
 R3     007000          4/60   11/23   12/55
 R4     011000          5/01   12/43   12/46   12/50
 R5     013000          5/02   11/39   11/43   11/60   12/32   12/36
 R6     015000          5/03
 R7     017000          5/04
 RAT1A  000023$ XD      1/07   10/08
 RAT2A  000024$ XD      1/07   10/10
 RAT3A  000025$ XD      1/07   10/12
 RAT4A  000026$ XD      1/07   10/14
 RAT5A  000027$ XD      1/07   10/16
 RAT6A  000030$ XD      1/07   10/18
 RAT7A  000031$ XD      1/07   10/20
 RATE   000350'         6/12    6/53    7/33    8/07    8/22
 RATE0  000620' XN      1/10    9/54
 RATE1  000621' XN      1/10    9/55
 RATE2  000622' XN      1/10    9/56
 RATE3  000623' XN      1/10    9/57
 RATE4  000624' XN      1/10    9/58
 RATE5  000625' XN      1/10    9/59
 RATE6  000626' XN      1/10    9/60
 RATE7  000627' XN      1/11   10/01
 RATEM  000617' XN      1/11    9/53
 REPT   000000  MC      5/32   11/18   11/19   11/21   11/22   11/23   11/26
                       11/27   11/29   11/33   11/35   11/37   11/38   11/39
                       11/40   11/42   11/43   11/47   11/49   11/50   11/51
                       11/52   11/55   11/56   11/57   12/01   12/02   12/03
                       12/05   12/06   12/07   12/08   12/10   12/17   12/20
                       12/21   12/22   12/26   12/30   12/31   12/33   12/34
                       12/35   12/39   12/40   12/41   12/42   12/47   12/48
                       12/52   12/53   12/54   12/55   12/56   12/58   12/60
                       13/01   13/02   13/03   13/04
 R.LN   000003$ XD      1/04    4/21    6/32
 R.PS   000005$ XD      1/04    8/38
 SP000  000234' EN      1/01    4/01    4/02    4/05   13/08
 SP001  000000' EN      1/19    1/20
 SP002  000340' EN      5/60    6/01    6/03
 SP010  000246'         4/18    4/36    4/45
 SP015  000265'         4/15    4/38
```

```
0009 MV1
007066'000403        JMP     INI40          ;SKIP IF ZERO
007067'157000        ADD     2,3            ;INDEX POINTER
007070'041400        STA     0,0,3          ;SAVE VALUE
              ;   ADDRESS MONITORING
007071'034011$INI40: LDA     3,MNADR        ;ADR OR ZERO
007072'156405        SUB     2,3,SNR        ;CHECK THIS ADR
007073'040010$       STA     0,MNVAL        ;SAVE VALUE FOR MONITORING
              ;RETURN
007074'126400        SUB     1,1            ; NO RESCHEDULING
007075'030403        LDA     2,INIA2
007076'034403        LDA     3,INIA3
007077'177777        .UIEX                  ;RETURN
007100'000000 INIA2: .-.
007101'000000 INIA3: .-.
007102'006555'AURT0: URT0
007103'036000 INALM: MKALM
007104'004000 ALMA:  1B4
007105'010000 ALMB:  1B3
              ;
              ;      CARD OUTPUT ROUTINE
              ;
                     .ENT    OTDCT
007106'007111'OTDCT: OTSAV            ;DCTSV - SAVE AREA ADR
007107'177777        -1               ;DCTMS - MASK
007110'007121'       OUTIS            ;DCTIS - INTERRUPT ADR
007111'000010 OTSAV: .BLK    8.
                     .ENT    OUTIS
007121'054760 OUTIS: STA     3,INIA3        ;SAVE RETURN ADR
007122'060257        NIOC    DCOUT
007123'020005-       LDA     0,INADR
007124'024757        LDA     1,INALM        ;CHECK IF ADR OR ALARM
007125'123405        AND     1,0,SNR
007126'000666        JMP     INIS+2         ;GO DO INPUT INTERRUPT
              ;
007127'004613        JSR     INST           ;OR RESTART INTERFACE
007130'126400        SUB     1,1            ;NO RESCHEDULING
007131'034750        LDA     3,INIA3
007132'007077'       .UIEX
                     .TITL   MV1
                     .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
 0010 MV1

ADR    000004- EN    1/22     1/27
ADRIS  006374'       2/34     3/46
ALMA   007104'       8/40     9/17
ALMB   007105'       8/52     9/18
AURT0  007102'       8/14     9/15
D10    006453'       3/34     3/37
DCIN   000056        1/02     7/32     8/20
DCOUT  000057        1/03     7/30     8/09     8/41     8/46     8/53     8/55
                     9/29
HIAD   001000        1/04     1/52     1/56     1/59     2/02     2/04     2/22
                     2/25
HIADR  000006- EN    1/22     1/29
IAD1   000021        1/06     3/06     3/15
IAD2   000022        1/07     3/08     3/19
IAD3   000023        1/08     3/10     3/20     3/23
INADR  000005- EN    1/22     1/28     6/57     7/24     7/33     8/17     9/30
INALM  007103'       7/29     9/16     9/31
INDCT  006777' EN    8/01     8/02
INI10  007045'       8/32     8/40
INI20  007063'       8/38     8/50     8/57
INI40  007071'       9/01     9/05
INIA2  007100'       8/10     9/10     9/13
INIA3  007101'       8/08     9/11     9/14     9/28     9/37
INIS   007012' EN    8/04     8/07     8/08     9/33
INS10  006750'       7/22     7/27
INS20  006772'       7/44     7/48
INS80  006773'       7/39     7/50
INSA0  006775'       7/27     7/50     7/53
INSA3  006776'       7/19     7/25     7/51     7/54
INSAV  007002'       8/32     8/05
INST   006742' EN    6/59     7/09     7/10     7/17     7/19     8/57     9/35
IN.ST  000025- EN    3/02     7/10     7/17
KM7    006375'       2/35     3/12
LGMNB  000003$ XD    2/28     5/42
LGMNC  000001$ XD    2/28     3/26     5/49     5/51
LGMNS  000002$ XD    2/28     5/45
LGMSA  006740'       5/52     7/07
LGMSG  006740' XN    7/07     7/11
LGSAV  000012- EN    1/30     1/34     3/28     8/59
MKADR  000010- EN    1/30     1/32     8/25
MKALM  036000        1/05     9/16
MNADR  000011$ XD    7/56     9/05
MNBAD  000012$ XD    7/56     8/48
MNMSA  006741'       6/20     7/08
MNVAL  000010$ XD    5/03     6/15     6/19     7/08     9/07
MTBFL  010000  EN    1/49     1/50
MTBHL  000010  EN    1/49     1/51
MTC    000004$ XD    2/50     5/03     5/15     5/21     8/11     8/23
MTCS   000006$ XD    2/49     5/03     5/17     5/19
MTFLG  000014$ XD    7/56     8/43     8/45
MTHDR  000000' EN    1/49     1/53
MTHLN  053012  EN    1/49     1/52
MTMSA  006737'       5/23     7/06
MTN    000005$ XD    2/52     5/03     5/16
MTNS   000007$ XD    2/51     5/03     5/18     5/20
MTTK1  006737' XN    2/29     7/06
OTDCT  007106' EN    9/22     9/23
OTSAV  007111'       9/23     9/26
OUTIS  007121' EN    9/25     9/27     9/28
```

```
0011 MV1

RAT1   005370' EN     1/37    2/24    2/25
RAT1A  000013- EN     1/36    1/37    2/31
RAT2   004367' EN     1/38    2/21    2/22
RAT2A  000014- EN     1/36    1/38
RAT3   004106' EN     1/39    2/18    2/19
RAT3A  000015- EN     1/36    1/39
RAT4   004042' EN     1/40    2/15    2/16
RAT4A  000016- EN     1/36    1/40
RAT5   004020' EN     1/41    2/12    2/13
RAT5A  000017- EN     1/36    1/41
RAT6   004014' EN     1/42    2/09    2/10
RAT6A  000020- EN     1/36    1/42
RAT7   004012' EN     1/43    2/06    2/07
RAT7A  000021- EN     1/36    1/43
RATE0  000000  EN     1/09    1/12
RATE1  000001  EN     1/09    1/13    4/34
RATE2  000012  EN     1/09    1/14    1/31    4/35
RATE3  000144  EN     1/09    1/15    2/19    4/36
RATE4  000764  EN     1/10    1/16    2/16    4/37
RATE5  001750  EN     1/10    1/17    2/13    4/38
RATE6  011610  EN     1/10    1/18    2/10    4/39
RATE7  023420  EN     1/10    1/19    2/07
RATEM  042362  EN     1/09    1/11    2/07    2/10    2/13    2/16    2/19
RMCNT  000013$ XD     7/56    8/42
SP     000022- EN     2/56    3/56    3/58    4/10    4/14    4/20    7/20
                      7/35    7/48
SPUA3  006537'        4/09    4/18    4/23    4/25
SPURZ  006540' EN     2/57    3/56    4/21    4/26
SPUSH  006517' EN     2/60    3/55    3/56    3/60    4/07    5/33    6/03
                      6/13    6/29    6/36
STAA   000023-        3/59    4/13    7/42
STAE   006475'        4/02    4/16
STAK   006476' EN     3/56    3/59    4/02    4/03
S.PUS  000024- EN     3/56    3/60    6/43    6/46    6/54
TAB1   000012' EN     1/23    1/54    1/55
TAB1A  000000- EN     1/22    1/23
TAB1I  000003- EN     1/22    1/26
TAB3   001012' EN     1/24    1/57    1/58
TAB3A  000001- EN     1/22    1/24    8/28
TAB4   002012' EN     1/25    1/60    2/01
TAB4A  000002- EN     1/22    1/25    8/34
TABLG  003012' EN     1/33    2/03    2/04
TLGA   000011- EN     1/30    1/33    3/27
TNUM0  000007- EN     1/30    1/31
UA1    006566'        2/32    4/54    5/32
UA1A   006372'        2/32    3/07
UA2    006567'        4/55    6/02
UA3    006570'        4/56    6/12
UA4    006571'        4/57    6/28
UA5    006572'        4/58    6/35
UA6    006573'        4/59    6/42
UA7    006574'        4/60    6/45
UAC0   006376'        2/36    2/44    3/51
UAC1   006377'        2/37    2/45    3/52
UAC2   006400'        2/38    2/46    3/53
UAC3   006401'        2/39    2/47    3/54
UAD1A  006371'        2/31    3/05
UAZER  006541'        2/59    4/28    6/53
```

```
0012 MV1

UCL10 006617'        5/12    5/28
UCL15 006640'        5/44    5/49
UCL20 006647' EN     5/29    5/36    5/47    5/50    5/56    5/58
UCL30 006657'        5/59    6/08
UCL40 006675'        6/09    6/17    6/24
UCL50 006703'        6/25    6/31
UCL60 006711'        6/32    6/38
UCL70 006717'        6/39    6/45
UCL80 006730' EN     6/50    6/55    6/57
UCL90 006733'        7/01
UCLA3 006736'        5/07    7/02    7/05
UCLIS 006575' EN     2/34    5/01    5/02    5/07
UCLST 006402' EN     2/27    2/43    2/44
UCT   023420         4/33    4/34    4/35    4/36    4/37    4/38    4/39
UCT1  006545'        4/34    5/30
UCT2  006546'        4/35    5/60
UCT3  006547'        4/36    6/10
UCT4  006550'        4/37    6/26
UCT5  006551'        4/38    6/33
UCT6  006552'        4/39    6/40
UCT60 006553'        4/40    5/38
UCTWR 006554'        4/41    5/13
UMIN  006544'        4/31    5/40    6/48    6/52
URT0  006555' EN     2/55    4/43    4/44    5/08    5/24    6/06    9/15
URT1  006556'        2/33    4/45    5/28    5/31
URT1A 006373'        2/33    3/09
URT2  006557'        4/46    5/58    6/01
URT3  006560'        4/47    6/08    6/11
URT4  006561'        4/48    6/24    6/27
URT5  006562'        4/49    6/31    6/34
URT6  006563'        4/50    6/38    6/41
URT60 006564'        4/51    5/35    5/39
URTWR 006565'        4/52    5/11    5/14
UST10 006433'        3/15    3/22
UST20 006450' EN     3/30    3/31    3/43
UST30 006457'        3/39    3/42
UST50 006464'        3/36    3/40    3/45
.IXMT 006673' XN     2/29    5/25    5/54    6/21
.UCEX 006735' XN     2/29    7/03
.UIEX 007132' XN     7/57    9/12    9/38
```

```
0001 MV2   MACRO REV 06.50             12:19:07 10/31/80
                     .TITL   OCASC
                     .ENT    OCASC,OC.AS
              ;CONVERT BINARY TO 6 CHARACTERS OCTAL ASCII
              ;   AC0  BYTE POINTER, UPDATED
              ;   AC1  NUMBER TO BE CONVERTED
                     .ZREL
000000-000000'OC.AS: OCASC
                     .NREL
              ;
000000'054442 OCASC: STA     3,OCA03
000001'050440        STA     2,OCA02
000002'044436        STA     1,OCA01
              ;
000003'030443        LDA     2,C6            ;GET COUNT
000004'050437        STA     2,OCA05
000005'143000        ADD     2,0             ;START BYTE POINTER AT RIGHT
000006'040437        STA     0,OCA07
000007'040430        STA     0,OCA00         ;BYTE POINTER
              ;
000010'030437 OCAS1: LDA     2,B10
000011'006013-       JSR     @DIV.I          ;DIVIDE BY 8
000012'030535        LDA     2,C60           ;ADD ASCII 0
000013'143000        ADD     2,0             ; TO REMAINDER
000014'044430        STA     1,OCA06         ;SAVE REMAINING NUMBER
              ;
000015'014430        DSZ     OCA07           ;DECR BYTE POINTER
000016'034427        LDA     3,OCA07
000017'175222        MOVZR   3,3,SZC
000020'000403        JMP     OCAS2
              ;LEFT BYTE
000021'024531        LDA     1,RMASK         ;MASK FOR RIGHT BYTE
000022'101301        MOVS    0,0,SKP         ;SWAP BYTES
              ;RIGHT BYTE
000023'024526 OCAS2: LDA     1,LMASK         ;MASK FOR LEFT BYTE
              ;COMMON PROCESSING
000024'031400        LDA     2,0,3           ;GET EXISTING WORD
000025'133400        AND     1,2             ;KEEP OTHER BYTE
000026'143000        ADD     2,0             ;COMBINE
000027'041400        STA     0,0,3           ;SAVE
000030'024414        LDA     1,OCA06         ;RETRIEVE REMAINING NUMBER
000031'014412        DSZ     OCA05           ;DECREMENT COUNT
000032'000756        JMP     OCAS1
              ;END PROCESSING
000033'020404        LDA     0,OCA00
000034'024404        LDA     1,OCA01
000035'030404        LDA     2,OCA02
000036'002404        JMP     @OCA03
              ;
000037'000000 OCA00: .-.
000040'000000 OCA01: .-.
000041'000000 OCA02: .-.
000042'000000 OCA03: .-.     ;RETURN ADDRESS
000043'000000 OCA05:.-.      ;COUNT
000044'000000 OCA06: .-.     ;SAVE FOR INTERMEDIATE
000045'000000 OCA07: .-.     ;BYTE POINTER
000046'000006 C6:    6
000047'000010 B10:   10
                     .TITL   BNASC
                     .ENT    BNASC,BN.AS
              ;CONVERT BINARY TO 5 CHARACTERS DECIMAL ASCII PLUS SIGN
```

```
0002 MV2
              ;   AC0  BYTE POINTER, UPDATED
              ;   AC1  NUMBER TO BE CONVERTED
                     .ZREL
000001-000050'BN.AS: BNASC
                     .NREL
              ;
000050'054467 BNASC: STA     3,BNA03
000051'050465        STA     2,BNA02
000052'044463        STA     1,BNA01
              ;
000053'030473        LDA     2,C5              ;GET COUNT
000054'050465        STA     2,BNA05
000055'143000        ADD     2,0               ;START BYTE POINTER AT RIGHT
000056'040465        STA     0,BNA07
000057'101400        INC     0,0               ;RETURN UPDATED POINTER
000060'040454        STA     0,BNA00           ;BYTE POINTER
              ;
000061'102400        SUB     0,0
000062'040456        STA     0,BNA04           ;NEGATIVE INDICATOR
000063'125133        MOVZL#  1,1,SNC
000064'000403        JMP     BNAS1
              ;
000065'124400        NEG     1,1
000066'044452        STA     1,BNA04           ;NEGATIVE INDICATOR
              ;
000067'030461 BNAS1: LDA     2,D10
000070'006013-       JSR     @DIV.I            ;DIVIDE BY 10
000071'030456        LDA     2,C60             ;ADD ASCII 0
000072'143000        ADD     2,0               ; TO REMAINDER
000073'044447        STA     1,BNA06           ;SAVE REMAINING NUMBER
              ;
000074'034447        LDA     3,BNA07
000075'175222        MOVZR   3,3,SZC
000076'000403        JMP     BNAS2
              ;LEFT BYTE
000077'024453        LDA     1,RMASK           ;MASK FOR RIGHT BYTE
000100'101301        MOVS    0,0,SKP           ;SWAP BYTES
              ;RIGHT BYTE
000101'024450 BNAS2: LDA     1,LMASK           ;MASK FOR LEFT BYTE
              ;COMMON PROCESSING
000102'031400        LDA     2,0,3             ;GET EXISTING WORD
000103'133400        AND     1,2               ;KEEP OTHER BYTE
000104'143000        ADD     2,0               ;COMBINE
000105'041400        STA     0,0,3             ;SAVE
000106'024434        LDA     1,BNA06           ;RETRIEVE REMAINING NUMBER
000107'014434        DSZ     BNA07             ;DECREMENT BYTE POINTER
000110'014431        DSZ     BNA05             ;DECREMENT COUNT
000111'000756        JMP     BNAS1
              ;END PROCESSING
000112'020432        LDA     0,CRBL            ;BLANK FOR + NUMBER
000113'024425        LDA     1,BNA04
000114'125004        MOV     1,1,SZR
000115'020430        LDA     0,CRMIN           ;MINUS FOR - NUMBER
000116'034425        LDA     3,BNA07
000117'175222        MOVZR   3,3,SZC
000120'000403        JMP     BNAS4
              ;LEFT BYTE
000121'024431        LDA     1,RMASK
000122'101301        MOVS    0,0,SKP
              ;RIGHT BYTE
```

```
000123'024426 BNAS4: LDA      1,LMASK
                     ;COMMON
000124'031400        LDA      2,0,3
000125'133400        AND      1,2
000126'143000        ADD      2,0
000127'041400        STA      0,0,3
                     ;
000130'020404        LDA      0,BNA00
000131'024404        LDA      1,BNA01
000132'030404        LDA      2,BNA02
000133'002404        JMP      @BNA03
                     ;
000134'000000 BNA00: .-.
000135'000000 BNA01: .-.
000136'000000 BNA02: .-.
000137'000000 BNA03: .-.      ;RETURN ADDRESS
000140'000000 BNA04: .-.      ;NEGATIVE INDICATOR
000141'000000 BNA05:.-.       ;COUNT
000142'000000 BNA06: .-.      ;SAVE FOR INTERMEDIATE
000143'000000 BNA07: .-.      ;BYTE POINTER
000144'000040 CRBL:  40
000145'000055 CRMIN: 55
000146'000005 C5:    5
000147'000060 C60:   60
000150'000012 D10:   10.
                     .TITL    GTCH
                     .ENT     GTCH,GT.CH
                     ;SUBROUTINE TO GET A CHARACTER
                     ;   AC0 - BYTE POINTER, UPDATED
                     ;   AC1 - CHARACTER, RETURNED
                     .ZREL
000002-000153'GT.CH: GTCH
                     .NREL
000151'177400 LMASK: 377B7
000152'000377 RMASK: 377
000153'054413 GTCH:  STA      3,GTCH3
000154'050411        STA      2,GTCH2
000155'040407        STA      0,GTCH0
                     ;
000156'004411        JSR      GTCR
                     ;
000157'105000        MOV      0,1
000160'020404        LDA      0,GTCH0
000161'101400        INC      0,0
000162'030403        LDA      2,GTCH2
000163'002403        JMP      @GTCH3
                     ;
000164'000000 GTCH0: .-.
000165'000000 GTCH2: .-.
000166'000000 GTCH3: .-.
                     ;
                     ;   SUBROUTINE TO GET CHAR
                     ;       AC0 - BYTE POINTER IN; CHARACTER OUT
000167'111222 GTCR:  MOVZR    0,2,SZC          ;DECIDE LEFT OR RIGHT CHAR
000170'000405        JMP      GTCR2            ;RIGHT
                     ;OBTAIN LEFT BYTE
000171'021000        LDA      0,0,2            ;GET WORD
000172'024757        LDA      1,LMASK          ;GET MASK
000173'123700        ANDS     1,0              ;ISOLATE BYTE
000174'000404        JMP      GTCR9
```

```
 0004 MV2
                     ;OBTAIN RIGHT BYTE
   000175'021000 GTCR2: LDA      0,0,2             ;GET WORD
   000176'024754        LDA      1,RMASK           ;GET MASK
   000177'123400        AND      1,0
                     ;RETURN CHARACTER
   000200'001400 GTCR9: JMP      0,3
                        .TITL    DIG
                        .ENT     DIG,DIG.
                     ;SUBROUTINE TO CONVERT ARBITRARY STRING OF ASCII DECIMAL
                     ;TO BINARY
                     ;   AC0  BYTE POINTER, UPDATED
                     ;   AC1  ANSWER OUT
                     ;   AC2  NUMBER OF DIGITS OUT
                     ;
                        .ZREL
   000003-000201'DIG.:  DIG
                        .NREL
   000201'054477 DIG:   STA      3,DIG03
   000202'040473        STA      0,DIG00
   000203'126400        SUB      1,1
   000204'044475        STA      1,DIG98
   000205'044475        STA      1,DIG99
                     ;OBTAIN BYTE
   000206'020467 DIG10: LDA      0,DIG00
   000207'004760        JSR      GTCR
                     ;DECODE CHARACTER
   000210'024737        LDA      1,C60
   000211'030472        LDA      2,C71             ;CHECK FOR DIGIT
   000212'142033        ADCZ#    2,0,SNC
   000213'106032        ADCZ#    0,1,SZC
   000214'000413        JMP      DIG90             ;END OF NUMBER
   000215'122400        SUB      1,0               ;GET VALUE
                     ;ADD TO ACCUMULATED SUM
   000216'024464        LDA      1,DIG99           ;GET SUM
   000217'131000        MOV      1,2               ;MULTIPLY BY 10
   000220'127120        ADDZL    1,1
   000221'147120        ADDZL    2,1
   000222'107000        ADD      0,1               ;ADD NEW DIGIT
   000223'044457        STA      1,DIG99
                     ;LOOP
   000224'010455        ISZ      DIG98             ;COUNT DIGITS
   000225'010450        ISZ      DIG00             ;INCR BYTE ADR
   000226'000760        JMP      DIG10
                     ;END OF DECODING
   000227'020446 DIG90: LDA      0,DIG00
   000230'024452        LDA      1,DIG99
   000231'030450        LDA      2,DIG98
   000232'002446        JMP      @DIG03
                     ;
                        .TITL    DIGN
                        .ENT     DIGN,DIG.N
                     ;SUBROUTINE TO CONVERT N DIGITS OF ASCII DECIMAL
                     ;TO BINARY
                     ;   AC0  BYTE POINTER, UPDATED
                     ;   AC1  N IN; ANSWER OUT
                     ;   RETURN+1  OK
                     ;   RETURN    ERROR
                     ;
                        .ZREL
   000004-000233'DIG.N: DIGN
```

```
0005 MV2
                            .NREL
000233'054445 DIGN:  STA     3,DIG03
000234'050443        STA     2,DIG02
000235'044441        STA     1,DIG01
000236'040437        STA     0,DIG00
              ;
000237'102400        SUB     0,0
000240'040442        STA     0,DIG99          ;ZERO ACCUMULATOR
000241'040440        STA     0,DIG98          ;ERROR COUNT
              ;
000242'020433 DIGN1: LDA     0,DIG00
000243'004724        JSR     GTCR
              ;DECODE CHARACTER
000244'024703        LDA     1,C60            ;CHECK FOR DIGIT
000245'030436        LDA     2,C71
000246'142033        ADCZ#   2,0,SNC
000247'106032        ADCZ#   0,1,SZC
000250'000422        JMP     DIGN9
000251'122400        SUB     1,0              ;GET VALUE
              ;ADD TO ACCUMULATED SUM
000252'024430 DIGN4: LDA     1,DIG99          ;GET SUM
000253'131000        MOV     1,2              ;MULTIPLY BY  10
000254'127120        ADDZL   1,1
000255'147120        ADDZL   2,1
000256'107000        ADD     0,1              ;ADD NEW DIGIT
000257'044423        STA     1,DIG99
              ;LOOP
000260'010415        ISZ     DIG00            ;INCR BYTE POINTER
000261'014415        DSZ     DIG01            ;DECR BYTE COUNT
000262'000760        JMP     DIGN1
              ;END OF DECODE
000263'020412        LDA     0,DIG00
000264'024416        LDA     1,DIG99
000265'030412        LDA     2,DIG02
000266'034413        LDA     3,DIG98
000267'175005        MOV     3,3,SNR          ;CHECK ERROR COUNT
000270'010410        ISZ     DIG03            ;IF ZERO, INCR RETURN
000271'002407        JMP     @DIG03
              ;ERROR ROUTINE
000272'010407 DIGN9: ISZ     DIG98
000273'102400        SUB     0,0
000274'000756        JMP     DIGN4
              ;
000275'000000 DIG00: .-.     ;BYTE POINTER
000276'000000 DIG01: .-.     ;SAVE AC1
000277'000000 DIG02: .-.     ;SAVE AC2
000300'000000 DIG03: .-.     ;RETURN ADR
000301'000000 DIG98: .-.     ;ERROR COUNT / DIGIT COUNT
000302'000000 DIG99: .-.     ;ANSWER
000303'000071 C71:   71
              ;
                     .TITL   RLN
                     .ENT    RLN,R.LN,IBUF,RDBF
       000002 MTTI=  2
       000003 MTTO=  3
              ;SUBROUTINE TO READ A LINE INTO RDBF
              ;   RETURN+1  OK
              ;   RETURN    CR ONLY
              ;
                      .ZREL
```

```
0006 MV2
000005-000404'R.LN:  RLN
000006-000610"IBUF:  RDBF*2
                     .NREL
000304'000100 RDBF:  .BLK    100
000404'040416 RLN:   STA     0,RLN00         ;SAVE ACCUMULATORS
000405'044416        STA     1,RLN01
000406'050416        STA     2,RLN02
000407'054416        STA     3,RLN03
              ;
000410'020006-       LDA     0,IBUF
000411'006017        .SYSTEM
000412'015402        .RDL    MTTI            ;READ LINE
000413'000400        JMP     .
              ;
000414'125234        MOVZR#  1,1,SZR         ;TEST FOR SINGLE CHARACTER
000415'010410        ISZ     RLN03           ;IF MORE, INCR RETURN ADR
              ;
000416'020404        LDA     0,RLN00
000417'024404        LDA     1,RLN01
000420'030404        LDA     2,RLN02
000421'002404        JMP     @RLN03
              ;
000422'000000 RLN00: .-.
000423'000000 RLN01: .-.
000424'000000 RLN02: .-.
000425'000000 RLN03: .-.
                     .TITL   PCR
                     .ENT    PCR,P.CR
              ;SUBROUTINE TO PRINT W/PAD AT START
              ;   AC0  BYTE POINTER
                     .ZREL
000007-000426'P.CR:  PCR
                     .NREL
000426!040412 PCR:   STA     0,PCR00
000427'044412        STA     1,PCR01
000430'050412        STA     2,PCR02
000431'054412        STA     3,PCR03
              ;
000432'020412        LDA     0,PCR04
000433'006017        .SYSTEM
000434'017003        .WRL    MTTO
000435'000400        JMP     .
              ;
000436'020402        LDA     0,PCR00
000437'000412        JMP     P.
              ;
000440'000000 PCR00: .-.
000441'000000 PCR01: .-.
000442'000000 PCR02: .-.
000443'000000 PCR03: .-.
000444'001112"PCR04: PCR05*2
000445'003000 PCR05: 3000    ;PAD,ZERO
                     .TITL   PNL
                     .ENT    PNL,P.NL
              ;SUBROUTINE TO PRINT W/ NO PADDING
              ;   AC0  BYTE POINTER
                     .ZREL
000010-000446'P.NL:  PNL
                     .NREL
000446'044773 PNL:   STA     1,PCR01
```

```
0007 MV2
000447'050773           STA     2,PCR02
000450'054773           STA     3,PCR03
                ;
000451'006017 P.:       .SYSTEM
000452'017003           .WRL    MTTO
000453'000400           JMP     .
                ;
000454'024765           LDA     1,PCR01
000455'030765           LDA     2,PCR02
000456'002765           JMP     @PCR03
                ;
                        .TITL   MULT
                        .ENT    MUL.A,MUL.U,MULU
                ;SUBROUTINE TO MULTIPLY
                        .ZREL
000011-000460'MUL.A:    MULA
000012-000457'MUL.U:    MULU
                        .NREL
000457'102460 MULU:     SUBC    0,0      ;CLEAR AC0, LEAVE CARRY
000460'054411 MULA:     STA     3,ML03   ;SAVE RETURN
000461'034411           LDA     3,ML20   ;GET STEP COUNT
                ;
000462'125203 ML99:     MOVR    1,1,SNC  ;CHECK NEXT BIT
000463'101201           MOVR    0,0,SKP  ;0 - SHIFT
000464'143220           ADDZR   2,0      ;1 - ADD AND SHIFT
000465'175404           INC     3,3,SZR  ;COUNT STEP
000466'000774           JMP     ML99     ;LOOP
                ;
000467'125260           MOVCR   1,1      ;SHIFT LAST LOW BIT
000470'002401           JMP     @ML03    ;RESTORE CARRY
                ;
000471'000000 ML03:     .-.
000472'177760 ML20:     -20      ;16 BITS
                        .TITL   DIVID
                        .ENT    DIV.I,DIV.U,DIVI,DIV.,DIVD
                ;SUBROUTINE TO DIVIDE
                        .ZREL
000013-000500'DIV.I:    DIVI
000014-000501'DIV.U:    DIVU
000015-000473'DIV.:     DIVD
                        .NREL
000473'054424 DIVD:     STA     3,DV03
000474'155220           MOVZR   2,3             ;GET HALF DIVISOR
000475'167022           ADDZ    3,1,SZC         ;FOR ROUNDING
000476'101400           INC     0,0
000477'000403           JMP     DIVU+1
000500'102400 DIVI:     SUB     0,0      ;INTEGER DIVIDE - CLEAR AC0
000501'054416 DIVU:     STA     3,DV03
000502'142432           SUBZ#   2,0,SZC  ;TEST OVERFLOW
000503'000412           JMP     DV99     ;YES - EXIT
000504'034766           LDA     3,ML20   ;GET STEP COUNT
000505'125120           MOVZL   1,1      ;SHIFT DIVIDEND LOW PART
                ;
000506'101100 DV98:     MOVL    0,0      ;SHIFT DIVIDEND HIGH PART
000507'142412           SUB#    2,0,SZC  ;DIVISOR IN?
000510'142400           SUB     2,0      ;YES
000511'125100           MOVL    1,1      ;SHIFT DIVIDEND LOW PART
000512'175404           INC     3,3,SZR  ;COUNT STEP
000513'000773           JMP     DV98     ;LOOP
                ;
```

```
0008 MV2
   000514'176441         SUBO    3,3,SKP ;DONE, CLEAR CARRY
   000515'176420 DV99:   SUBZ    3,3     ;SET CARRY
   000516'002401         JMP     @DV03   ;RETURN
                 ;
   000517'000000 DV03:   .-.
                         .TITL   MV2
                         .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
     0009 MV2

   B10    000047'        1/20    1/57
   BNA00 000134'         2/16    3/08    3/13
   BNA01 000135'         2/09    3/09    3/14
   BNA02 000136'         2/08    3/10    3/15
   BNA03 000137'         2/07    3/11    3/16
   BNA04 000140'         2/19    2/24    2/51    3/17
   BNA05 000141'         2/12    2/47    3/18
   BNA06 000142'         2/30    2/45    3/19
   BNA07 000143'         2/14    2/32    2/46    2/54    3/20
   BNAS1 000067'         2/21    2/26    2/48
   BNAS2 000101'         2/34    2/39
   BNAS4 000123'         2/56    3/01
   BNASC 000050' EN      1/58    1/59    2/04    2/07
   BN.AS 000001- EN      1/59    2/04
   C5     000146'        2/11    3/23
   C6     000046'        1/14    1/56
   C60    000147'        1/22    2/28    3/24    4/27    5/14
   C71    000303'        4/28    5/15    5/50
   CRBL   000144'        2/50    3/21
   CRMIN 000145'         2/53    3/22
   D10    000150'        2/26    3/25
   DIG    000201' EN     4/07    4/08    4/16    4/18
   DIG00 000275'         4/19    4/24    4/42    4/45    5/05    5/11    5/28
                         5/32    5/44
   DIG01 000276'         5/04    5/29    5/45
   DIG02 000277'         5/03    5/34    5/46
   DIG03 000300'         4/18    4/48    5/02    5/37    5/38    5/47
   DIG10 000206'         4/24    4/43
   DIG90 000227'         4/31    4/45
   DIG98 000301'         4/21    4/41    4/47    5/09    5/35    5/40    5/48
   DIG99 000302'         4/22    4/34    4/39    4/46    5/08    5/21    5/26
                         5/33    5/49
   DIGN   000233' EN     4/50    4/51    4/60    5/02
   DIGN1 000242'         5/11    5/30
   DIGN4 000252'         5/21    5/42
   DIGN9 000272'         5/18    5/40
   DIG.   000003- EN     4/08    4/16
   DIG.N 000004- EN      4/51    4/60
   DIVD   000473' EN     7/35    7/40    7/42
   DIVI   000500' EN     7/35    7/38    7/47
   DIVU   000501'        7/39    7/46    7/48
   DIV.   000015- EN     7/35    7/40
   DIV.I 000013- EN      1/21    2/27    7/35    7/38
   DIV.U 000014- EN      7/35    7/39
   DV03   000517'        7/42    7/48    8/03    8/05
   DV98   000506'        7/54    7/59
   DV99   000515'        7/50    8/02
   GTCH   000153' EN     3/26    3/27    3/32    3/36
   GTCH0 000164'         3/38    3/43    3/48
   GTCH2 000165'         3/37    3/45    3/49
   GTCH3 000166'         3/36    3/46    3/50
   GTCR   000167'        3/40    3/54    4/25    5/12
   GTCR2 000175'         3/55    4/02
   GTCR9 000200'         3/60    4/06
   GT.CH 000002- EN      3/27    3/32
   IBUF   000006- EN     5/53    6/02    6/10
   LMASK 000151'         1/34    2/39    3/01    3/34    3/58
   ML03   000471'        7/20    7/30    7/32
   ML20   000472'        7/21    7/33    7/51
```

```
 0010 MV2

ML99  000462'        7/23    7/27
MTTI  000002         5/54    6/12
MTTO  000003         5/55    6/41    7/05
MULA  000460'        7/16    7/20
MULU  000457' EN     7/13    7/17    7/19
MUL.A 000011- EN     7/13    7/16
MUL.U 000012- EN     7/13    7/17
OCA00 000037'        1/18    1/44    1/49
OCA01 000040'        1/12    1/45    1/50
OCA02 000041'        1/11    1/46    1/51
OCA03 000042'        1/10    1/47    1/52
OCA05 000043'        1/15    1/41    1/53
OCA06 000044'        1/24    1/40    1/54
OCA07 000045'        1/17    1/26    1/27    1/55
OCAS1 000010'        1/20    1/42
OCAS2 000023'        1/29    1/34
OCASC 000000' EN     1/01    1/02    1/07    1/10
OC.AS 000000- EN     1/02    1/07
PCR   000426' EN     6/27    6/28    6/32    6/34
PCR00 000440'        6/34    6/44    6/47
PCR01 000441'        6/35    6/48    6/60    7/08
PCR02 000442'        6/36    6/49    7/01    7/09
PCR03 000443'        6/37    6/50    7/02    7/10
PCR04 000444'        6/39    6/51
PCR05 000445'        6/51    6/52
PNL   000446' EN     6/53    6/54    6/58    6/60
P.    000451'        6/45    7/04
P.CR  000007- EN     6/28    6/32
P.NL  000010- EN     6/54    6/58
RDBF  000304' EN     5/53    6/02    6/04
RLN   000404' EN     5/52    5/53    6/01    6/05
RLN00 000422'        6/05    6/18    6/23
RLN01 000423'        6/06    6/19    6/24
RLN02 000424'        6/07    6/20    6/25
RLN03 000425'        6/08    6/16    6/21    6/26
RMASK 000152'        1/31    2/36    2/58    3/35    4/03
R.LN  000005- EN     5/53    6/01
```

```
                           .TITL   MV3     ;COMMON SUBS
                   ;
                           .EXTD   HIADR,TAB1A,ADR,IBUF
                           .EXTD   GT.CH,DIG.,P.CR
                   ;
                           .ZREL
                           .ENT    LADR1,LADR2,LINCR,IFMOD,ERCT,BFPT
000000-000000 LADR1:       .-.     ;START ADR
000001-000000 LADR2:       .-.     ;END ADR
000002-000000 LINCR:       .-.     ;INCREMENT
000003-000000 IFMOD:       .-.     ;DETERMINES IF MODS ALLOWED
000004-000000 ERCT:        .-.     ;ERROR COUNT, INITIALIZED HERE
000005-000000 BFPT:        .-.     ;TEMP STORAGE, BYTE POINTER
                   ;
                           .NREL
        000001             .TXTM   1
                           .ENT    MV003
                   MV003:
000000'040504 Q11:         .TXT    /ADR INVALID<15>/
        051040
        044516
        053101
        046111
        042015
        000000
000007'040504 Q12:         .TXT    /ADR OUT OF RANGE<15>/
        051040
        047525
        052040
        047506
        020122
        040516
        043505
        006400
000020'043117 Q13:         .TXT    /FORMAT ERROR<15>/
        051115
        040524
        020105
        051122
        047522
        006400
000027'040504 Q14:         .TXT    /ADR1 > ADR2<15>/
        051061
        020076
        020101
        042122
        031015
        000000
000036'044516 Q15:         .TXT    /INCREMENT INVALID<15>/
        041522
        042515
        042516
        052040
        044516
        053101
        046111
        042015
        000000
000050'041510 Q17:         .TXT    /CHECK MODE REQUIRES ADRS ONLY<15>/
        042503
```

```
0002 MV3
       045440
       046517
       042105
       020122
       042521
       052511
       051105
       051440
       040504
       051123
       020117
       047114
       054415
       000000
               ;
                       .ENT    Q12.2,Q13.2
       000016"Q12.2= Q12*2
       000040"Q13.2= Q13*2
000070'000000"Q11B:    Q11*2
000071'000016"Q12B:    Q12*2
000072'000056"Q14B:    Q14*2
000073'000074"Q15B:    Q15*2
000074'000120"Q17B:    Q17*2
                       .TITL   GADR
                       .ENT    GADR,G.ADR
               ;DECODE ADR, ADR-ADR, OR ADR-ADR/INCR
               ;   AC0    - BYTE POINTER, UPDATED
               ;   RETURN- DONE, STATUS REPORTED
               ;          - ERROR: COMMA IN CHECK MODE
               ;   RETURN+1-GO ON: COMMA, MODIFY MODE
               ;
                       .ZREL
000006-000075'G.ADR:   GADR
                       .NREL
000075'054477 GADR:    STA     3,GADA3           ;SAVE RETURN ADDRESS
000076'102400          SUB     0,0
000077'040004-         STA     0,ERCT            ;ZERO ERROR COUNT
000100'020004$         LDA     0,IBUF            ;GET BYTE POINTER
               ;GET FIRST ADR
000101'004500          JSR     GTAD
000102'044000-         STA     1,LADR1           ;SAVE START ADR
000103'044001-         STA     1,LADR2           ;END SAME BY DEFAULT
000104'126520          SUBZL   1,1
000105'044002-         STA     1,LINCR           ;INCR 1 BY DEFAULT
               ;CHECK FOR DASH
000106'006005$         JSR     @GT.CH
000107'030466          LDA     2,QRDS
000110'132404          SUB     1,2,SZR
000111'000431          JMP     GAD40             ;NO - CHECK CR OR COMMA
               ;GET NEXT ADR
000112'004467          JSR     GTAD
000113'044001-         STA     1,LADR2           ;SAVE END ADR
               ;CHECK ADR1 <= ADR2
000114'030000-         LDA     2,LADR1
000115'146432          SUBZ#   2,1,SZC
000116'000406          JMP     GAD10
               ;    ERROR - ADR2<ADR1
000117'040005-         STA     0,BFPT
000120'020752          LDA     0,Q14B
000121'006007$         JSR     @P.CR
```

```
0003 MV3
000122'010004-       ISZ     ERCT
000123'020005-       LDA     0,BFPT
              ;CHECK FOR SLASH
000124'006005$GAD10: JSR     @GT.CH
000125'030451        LDA     2,QRSL
000126'132404        SUB     1,2,SZR
000127'000413        JMP     GAD40              ;NO - CHECK CR OR COMMA
              ;GET INCR
000130'006006$       JSR     @DIG.
000131'044002-       STA     1,LINCR
000132'151004        MOV     2,2,SZR
000133'000406        JMP     GAD30
              ;   ERROR - INCR INVALID
000134'040005-       STA     0,BFPT
000135'020736        LDA     0,Q15B
000136'006007$       JSR     @P.CR
000137'010004-       ISZ     ERCT
000140'020005-       LDA     0,BFPT
              ;GET NEXT CHAR
000141'006005$GAD30: JSR     @GT.CH
              ;CHECK CR OR COMMA
000142'030435 GAD40: LDA     2,QRCR
000143'132404        SUB     1,2,SZR
000144'000415        JMP     GAD60
              ;CR - REPORT STATUS LOOP
000145'024004-       LDA     1,ERCT
000146'125004        MOV     1,1,SZR
000147'000424        JMP     GAD90              ;RETURN IF ERROR
              ;   LOOP
000150'020002-       LDA     0,LINCR
000151'024000-       LDA     1,LADR1
000152'030001-       LDA     2,LADR2
000153'132433 GAD50: SUBZ#   1,2,SNC
000154'000417        JMP     GAD90              ;LOOP END
              ;
000155'044003$       STA     1,ADR
000156'006007-       JSR     @R.PS
000157'107000        ADD     0,1
000160'000773        JMP     GAD50
              ;IF CHECK MODE, MUST BE CR
000161'030003-GAD60: LDA     2,IFMOD
000162'151004        MOV     2,2,SZR
000163'000404        JMP     GAD70              ;MAY BE COMMA
000164'020710        LDA     0,Q17B             ;TYPE "ONLY ADR REQ'D"
000165'006007$       JSR     @P.CR
000166'000405        JMP     GAD90
              ;   ELSE, MAY BE COMMA
000167'030411 GAD70: LDA     2,QRCM
000170'132404        SUB     1,2,SZR
000171'006011-       JSR     @FM.ER             ;FORMAT ERROR
              ;RETURN
000172'010402        ISZ     GADA3              ;RETURN+1
000173'002401 GAD90: JMP     @GADA3             ;RETURN
              ;
000174'000000 GADA3: .-.
000175'000055 QRDS:  "-
000176'000057 QRSL:  "/
000177'000015 QRCR:  15
000200'000054 QRCM:  ",
                     .TITL   GADR
```

```
                       .ENT    GTAD
              ;
              ;   GET & CHECK A SINGLE ADR
              ;
000201'054422 GTAD:    STA     3,GTAA3
000202'006006$         JSR     @DIG.
000203'151004          MOV     2,2,SZR
000204'000404          JMP     GTA10
              ;TYPE "ADR INVALID"
000205'040005-         STA     0,BFPT
000206'020662          LDA     0,Q11B
000207'000410          JMP     GTA30
              ;CHECK RANGE
000210'125005 GTA10:   MOV     1,1,SNR
000211'000404          JMP     GTA20
000212'030001$         LDA     2,HIADR
000213'146433          SUBZ#   2,1,SNC
000214'000406          JMP     GTA90
              ;TYPE "ADR OUT OF RANGE"
000215'040005-GTA20:   STA     0,BFPT
000216'020653          LDA     0,Q12B
000217'006007$GTA30:   JSR     @P.CR
000220'010004-         ISZ     ERCT
000221'020005-         LDA     0,BFPT
              ;RETURN ADR IN AC1
000222'002401 GTA90:   JMP     @GTAA3
000223'000000 GTAA3:   .-.
                       .TITL   RPS     ;REPORT PRESENT STATUS
                       .ENT    R.PS,RPS
                       .EXTD   TAB1A,TAB3A,TAB4A,ADR
                       .EXTD   P.CR,P.NL,BN.AS
              ;SUBROUTINE TO REPORT PRESENT STATUS OF AN ADR
              ;        ADR IN ADR
                       .ZREL
000007-000255'R.PS:    RPS
                       .NREL
       000001          .TXTM   1
000224'054130 Q00:     .TXT    /XXXXXX,X,X/
       054130
       054130
       026130
       026130
       000000
000232'026000 Q01:     .TXT    /,/
000233'054130 Q02:     .TXT    /XXXXXX/
       054130
       054130
       000000
000237'006400 Q03:     .TXT    /<15>/
              ;
000240'000450"Q00B:    Q00*2
000241'000453"Q00B3:   Q00*2+3
000242'000464"Q01B:    Q01*2
000243'000466"Q02B:    Q02*2
000244'000470"Q02B2:   Q02*2+2
000245'000476"Q03B:    Q03*2
              ;
000246'026122 QLR:     26122
000247'026120 QLP:     26120
000250'026060 QB0CM:   26060
```

```
 0005 MV3
000251'001000 MKMOD: 1B6
000252'016000 MKRAT: 7B5
000253'001777 D1023: 1023.
000254'000000 MAX:    .-.
                      .TITL    RPS
000255'054473 RPS:    STA      3,RPSA3
000256'050471         STA      2,RPSA2
000257'044467         STA      1,RPSA1
000260'040465         STA      0,RPSA0
              ;CONVERT ADR
000261'024003$        LDA      1,ADR
000262'020756         LDA      0,Q00B
000263'006013$        JSR      @BN.AS
              ;    GET STATUS WORD
000264'034002$        LDA      3,TAB1A
000265'137000         ADD      1,3
000266'031400         LDA      2,0,3          ;GET STATUS WORD
              ;CONVERT RATE
000267'024763         LDA      1,MKRAT        ;MASK FOR RATE
000270'147700         ANDS     2,1
000271'125220         MOVZR    1,1            ;MOVE TO BITS 5-7
000272'125220         MOVZR    1,1
000273'020755         LDA      0,QB0CM        ;ASCII MASK
000274'107000         ADD      0,1
000275'044732         STA      1,Q00+3        ;SAVE IN PRINT BUFFER
              ;CONVERT MODE
000276'024753         LDA      1,MKMOD        ;MASK FOR MODE
000277'020750         LDA      0,QLP          ;ZERO MEANS PEAK
000300'147404         AND      2,1,SZR        ;CHECK BIT
000301'020745         LDA      0,QLR          ;ONE MEANS REAL
000302'040726         STA      0,Q00+4        ;SAVE IN PRINT BUFFER
              ;PRINT
000303'020736         LDA      0,Q00B3        ;PRINT ADR,RATE & MODE
000304'006007$        JSR      @P.CR
              ;
              ;    GET MIN & MAX
000305'030003$        LDA      2,ADR
000306'034010$        LDA      3,TAB3A
000307'157000         ADD      2,3
000310'025400         LDA      1,0,3
000311'034011$        LDA      3,TAB4A
000312'157000         ADD      2,3
000313'021400         LDA      0,0,3
000314'040740         STA      0,MAX
              ;    CHECK FOR NOT DEFAULT
000315'125004         MOV      1,1,SZR
000316'000404         JMP      RPS10
000317'030734         LDA      2,D1023
000320'112405         SUB      0,2,SNR
000321'000416         JMP      RPS90
              ;
000322'020720 RPS10:  LDA      0,Q01B         ;PRINT COMMA
000323'006012$        JSR      @P.NL
              ;CONVERT MINIMUM
000324'020717         LDA      0,Q02B         ;BYTE POINTER
000325'006013$        JSR      @BN.AS         ;CONVERT TO ASCII
              ;
000326'020716         LDA      0,Q02B2        ;PRINT LAST 4
000327'006012$        JSR      @P.NL
000330'020712         LDA      0,Q01B         ;PRINT COMMA
```

```
000331'006012$        JSR     @P.NL
               ;CONVERT MAXIMUM
000332'024722         LDA     1,MAX
000333'020710         LDA     0,Q02B
000334'006013$        JSR     @BN.AS
               ;
000335'020707         LDA     0,Q02B2
000336'006012$        JSR     @P.NL
               ;PRINT LINE
000337'020706 RPS90:  LDA     0,Q03B          ;PRINT CR
000340'006012$        JSR     @P.NL           ;PRINT PRESENT STATUS
               ;
000341'020404         LDA     0,RPSA0
000342'024404         LDA     1,RPSA1
000343'030404         LDA     2,RPSA2
000344'002404         JMP     @RPSA3
               ;
000345'000000 RPSA0:  .-.
000346'000000 RPSA1:  .-.
000347'000000 RPSA2:  .-.
000350'000000 RPSA3:  .-.
                      .TITL   CKCR
                      .ENT    CKCR,CK.CR
                      .EXTD   GT.CH
               ;GET & CHECK CR AT END OF LINE
               ;   AC0 - BYTE POINTER, UPDATED
               ;   AC1 - CHARACTER RETURNED IF NOT CR
               ;   RETURN   - NO CR
               ;   RETURN+1 - CR
                      .ZREL
000010-000351'CK.CR:  CKCR
                      .NREL
000351'054415 CKCR:   STA     3,CKCA3
000352'050415         STA     2,CKCA2
000353'044415         STA     1,CKCA1
000354'006005$        JSR     @GT.CH
000355'030622         LDA     2,QRCR
000356'132405         SUB     1,2,SNR
000357'000403         JMP     CKC80
               ;   NOT CR
000360'004411         JSR     FMTER
000361'000403         JMP     CKC90
               ;   RETURN
000362'010404 CKC80:  ISZ     CKCA3
000363'024405         LDA     1,CKCA1
000364'030403 CKC90:  LDA     2,CKCA2
000365'002401         JMP     @CKCA3
               ;
000366'000000 CKCA3:  .-.
000367'000000 CKCA2:  .-.
000370'000000 CKCA1:  .-.
                      .TITL   FMTER
                      .ENT    FMTER,FM.ER
                      .EXTD   P.CR
               ;PRINT "FORMAT ERROR" AND INCR ERCT
                      .ZREL
000011-000371'FM.ER:  FMTER
                      .NREL
000371'054407 FMTER:  STA     3,FMTA3
000372'040005-        STA     0,BFPT
```

```
 0016 SP000

SP020 000271'       4/28    4/43
SP070 000305' EN    5/17    5/18    5/20    5/28
SP080 000316' EN    4/32    5/29    5/39    5/40
SP090 000322'       4/41    5/45
SP091 000327'       5/46    5/51
SP093 000330'       4/22    5/52    6/33
SP0A0 000331'       4/05    5/53   11/12
SP0A1 000332'       4/06    5/54
SP0A2 000333'       4/07    5/55
SP0A3 000334'       4/08    5/56
SP100 000364' EN    5/47    6/26    6/27    6/29    6/36    7/29    8/41
SP120 000406'       6/44    6/52
SP130 000415'       6/55    7/01
SP140 000416'       6/58    7/02
SP150 000435'       7/09    7/13    7/21
SP160 000440' EN    6/59    7/26    7/27
SP164 000460'       7/40    7/47
SP170 000470' EN    7/35    7/50    7/57    7/58    8/36
SP180 000530'       8/14    8/38
SP190 000531'       7/60    8/40
SP400 000537' EN    8/13    8/47    8/48    8/52
SP410 000555'       8/59    9/11
SP420 000563'       9/13    9/17
SP430 000567'       9/15    9/22
SP450 000600' EN    5/51    8/40    9/32    9/33    9/35
SP4A0 000574'       8/52    9/22    9/28
SP4A1 000575'       8/53    9/23    9/29
SP4A2 000576'       8/54    9/24    9/30
SP4A3 000577'       8/55    9/17    9/25    9/31    9/35    9/46
SP800 000631' EN    5/52   10/03   10/04   10/08
SP820 000654'      10/29   10/44
SP830 000670'      10/32   10/42
SP891 000717'      10/25   10/27   10/43   11/01
SP900 000720'      10/56   11/05
SP990 000726'      11/05   11/12
T0    000620'       9/54   10/02
TAB1A 000015$ XD    1/06    5/50    8/03   10/23
TAB1I 000016$ XD    1/06    8/05    8/08    8/30
TABA  000727' EN    5/49   11/15   11/16   11/17
TABAA 000325'       5/09    5/49
TABQ  000326'       5/10    5/50
TMAXM 000617' EN    9/11    9/37    9/52    9/53
TMT1  000707'      10/09   10/11   10/13   10/15   10/17   10/19   10/21
                   10/59   10/60
TMT1A 000706'      10/35   10/47   10/59
TNUM  000000- EN    1/16    1/17    5/14  _ 8/57    9/14    9/38
TNUM0 000040$ XD    1/08
TNUMA 000352'       5/13    6/14
VB1SB 000335'       4/17    5/57
VB200 000335' XN    1/11    5/57
VB220 000336' XN    1/11    5/58
VB240 000337' XN    1/11    5/59
VB2SB 000336'       5/43    5/58
VB3SB 000337'       4/39    5/59
WRN.A 000012$ XD    1/05    4/44
Z0    000000        5/05   11/20   11/22   11/24   11/25   11/29   11/30
                   11/31   11/34   11/36   11/41   11/44   11/46   11/48
                   11/51   11/53   11/54   11/57   11/58   12/03   12/04
```

```
0017 SP000

                  12/07    12/09    12/11    12/13    12/15    12/18    12/19
                  12/22    12/23    12/25    12/27    12/29    12/34    12/36
                  12/38    12/41    12/43    12/49    12/53    12/56    12/57
                  12/59    13/02    13/04    13/05
```

```
0001 AL000 MACRO REV 06.50        -        19:48:01 10/31/80
                        .TITL   AL000   ;STORAGE
              ;
                        .EXTN   VB300,VB320,VB240
                        .EXTN   Q01.2,Q07.2,Q08.2
              ;
                        .EXTD   DIG.,ĠT.CH,R.LN,P.CR,FM.ER,CK.CR
                        .EXTD   IBUF,MVINI,WRN.A,G.ADR,R.PS
                        .EXTD   TAB1A,TAB3A,TAB4A,ADR,HIADR
                        .EXTD   IFMOD,LADR1,LADR2,LINCR,BFPT,ERCT
              ;
       000001           .TXTM   1
              ;
                        .NREL
                        .ENT    AL001
              AL001:
000000'005040 Q00:      .TXT    /<12> * * * ALARM LIMITS * * *<15>/
       025040
       025040
       025040
       040514
       040522
       046440
       046111
       046511
       052123
       020052
       020052
       020052
       006400
000016'041517 Q02:      .TXT    /CONTINUING DESTROYS EXISTING LIMITS, /
       047124
       044516
       052511
       047107
       020104
       042523
       052122
       047531
       051440
       042530
       044523
       052111
       047107
       020114
       044515
       044524
       051454
       020000
000041'042516 Q06:      .TXT    /ENTER ADRS OR ADRS,MIN,MAX:<15>/
       052105
       051040
       040504
       051123
       020117
       051040
       040504
       051123
       026115
       044516
       026115
```

```
              040530
              035015
              000000
       000060'046511 Q14:   .TXT    /MIN INVALID<15>/
              047040
              044516
              053101
              046111
              042015
              000000
       000067'046501 Q15:   .TXT    /MAX INVALID<15>/
              054040
              044516
              053101
              046111
              042015
              000000
       000076'046511 Q16:   .TXT    /MIN > MAX<15>/
              047040
              037040
              046501
              054015
              000000
                            .TITL   AL000   ;PRELIMINARY
                            .ENT    AL000
                     ;
                     ; * * * ALARM LIMIT SELECTION * * *
                     ;
000104'040463 AL000: STA    0,AL0A0
000105'044463        STA    1,AL0A1
000106'050463        STA    2,AL0A2
000107'054463        STA    3,AL0A3
                     ;
000110'020475        LDA    0,Q00B
000111'006004$       JSR    @P.CR            ;TYPE "ALARM LIMIT"
                     ;
000112'020010$       LDA    0,MVINI
000113'101005        MOV    0,0,SNR
000114'000421        JMP    AL015            ;CHECK MODE ONLY
                     ;ASK "INIT,MOD, OR CK?"
000115'006456        JSR    @VB1SB           ;EXPLAIN INIT,MOD,CK
000116'020470 AL010: LDA    0,Q01B
000117'006004$       JSR    @P.CR            ;TYPE "INIT,MOD OR CK?"
                     ;
000120'006003$       JSR    @R.LN            ;READ A LINE
000121'002475        JMP    @AL093           ;JUST CR, GO TO NEXT SEC
000122'020007$       LDA    0,IBUF
000123'006002$       JSR    @GT.CH           ;GET CHARACTER
                     ;
000124'020452        LDA    0,QRI
000125'122405        SUB    1,0,SNR
000126'000413        JMP    AL020            ;INITIALIZE
                     ;
000127'020450        LDA    0,QRM
000130'122405        SUB    1,0,SNR
000131'000427        JMP    AL080            ;MODOFY
                     ;
000132'020446        LDA    0,QRC
000133'122404        SUB    1,0,SZR
000134'000762        JMP    AL010            ;TRY AGAIN
```

```
                          ;CHECK ONLY - DON'T HALT PROCESSING
000135'040021$AL015: STA      0,IFMOD            ;ZERO FLAG FOR NO MODS
000136'006437        JSR      @VB3SB             ;EXPLAIN ADRS
000137'020452        LDA      0,Q07B             ;TYPE "ENTER ADR"
000140'000424        JMP      AL090
                   ;INITIALIZE
000141'020446 AL020: LDA      0,Q02B             ;WARNING MESSAGE
000142'006011$       JSR      @WRN.A             ;CHANCE TO ABORT
000143'000753        JMP      AL010              ;ABORT
                   ;    MIN & MAX TABLES
000144'024020$       LDA      1,HIADR
000145'044024$       STA      1,LINCR
000146'030015$       LDA      2,TAB3A
000147'021000        LDA      0,0,2              ;DEFAULT MIN
000150'034016$       LDA      3,TAB4A
000151'025400        LDA      1,0,3              ;DEFAULT MAX
                   ;
000152'041000 AL040: STA      0,0,2
000153'045400        STA      1,0,3
000154'151400        INC      2,2
000155'175400        INC      3,3
000156'014024$       DSZ      LINCR
000157'000773        JMP      AL040
                   ;INIT OR MODIFY
000160'102520 AL080: SUBZL    0,0                ;GET CONSTANT 1
000161'040021$       STA      0,IFMOD            ;SET FLAG TO PERMIT MODS
                   ;                             ;EXPLAIN AL SELECTION
000162'006412        JSR      @VB2SB             ;TYPE "ENTER ADR,MIN,MAX"
000163'020425        LDA      0,Q06B
000164'006004$AL090: JSR      @P.CR
000165'105000        MOV      0,1
000166'000431        JMP      AL100
000167'000000 AL0A0: .-.
000170'000000 AL0A1: .-.
000171'000000 AL0A2: .-.
000172'000000 AL0A3: .-.
000173'177777 VB1SB: VB300
000174'177777 VB2SB: VB320
000175'177777 VB3SB: VB240
                     .TITL    AL002    ;STORAGE
                     .ENT     AL002
                   ;CONSTANTS & STORAGE
                   AL002:
000176'000111 QRI:   "I
000177'000115 QRM:   "M
000200'000103 QRC:   "C
000201'000054 QRCM:  ",
000202'001777 K1777: 1777     ;MAXIMUM RETURNED VALUE
000203'000000 MIN:   .-.
000204'000000 MAX:   .-.
                   ;
000205'000000"Q00B:  Q00*2
000206'177777 Q01B:  Q01.2
000207'000034"Q02B:  Q02*2
000210'000102"Q06B:  Q06*2
000211'177777 Q07B:  Q07.2
000212'177777 Q08B:  Q08.2
000213'000140"Q14B:  Q14*2
000214'000156"Q15B:  Q15*2
000215'000174"Q16B:  Q16*2
```

```
0004 AL000
                 ;
000216'000357'AL093: AL900
                     .TITL    AL100    ;PROCESS
                     .ENT     AL100
                 ;ROUTINE TO LOOP FOR ALARM LIMIT MODIFICATION
000217'020773 AL100: LDA      0,Q08B                    ;ISSUE PROMPT
000220'006004$       JSR      @P.CR
                 ;READ LINE - INITIALIZE
000221'006003$       JSR      @R.LN
000222'002774        JMP      @AL093
                 ;GET & CHECK ADR(S)
000223'006012$       JSR      @G.ADR
000224'000773        JMP      AL100                     ;ERROR OR REQUEST
000225'126400        SUB      1,1
000226'044530        STA      1,IFDEF                   ;INITIALIZE FOR DEFAULT
                 ;GET & CHECK MIN
000227'006001$       JSR      @DIG.                     ;CONVERT DIGIT STRING
000230'044753        STA      1,MIN                     ;SAVE VALUE
000231'151004        MOV      2,2,SZR                   ;CHECK FOR NO DIGITS
000232'000405        JMP      AL120
                 ;   NO MIN: USE PRESENT VALUE
000233'126000        ADC      1,1
000234'044522        STA      1,IFDEF
000235'044746        STA      1,MIN
000236'000411        JMP      AL130
                 ;   CHECK VALUE                        ;MAX IS 1023
000237'030743 AL120: LDA      2,K1777
000240'132432        SUBZ#    1,2,SZC                   ;OK - AC1 <= 1023
000241'000406        JMP      AL130
                 ;   ERROR
000242'040025$       STA      0,BFPT
000243'020750        LDA      0,Q14B                    ;TYPE "MIN INVALID"
000244'006004$       JSR      @P.CR
000245'010026$       ISZ      ERCT
000246'020025$       LDA      0,BFPT
                 ;
                 ;GET & CHECK COMMA
000247'006002$AL130: JSR      @GT.CH
000250'030731        LDA      2,QRCM
000251'132404        SUB      1,2,SZR                   ;TYPE "FORMAT ERROR"
000252'006005$       JSR      @FM.ER
                 ;GET & CHECK MAX
000253'006001$       JSR      @DIG.
000254'044730        STA      1,MAX
000255'151004        MOV      2,2,SZR
000256'000405        JMP      AL140
                 ;   NO MAX: USE PRESENT VALUE
000257'126000        ADC      1,1
000260'044476        STA      1,IFDEF
000261'044723        STA      1,MAX
000262'000411        JMP      AL150
                 ;   CHECK VALUE                        ;MAX IS 1023
000263'030717 AL140: LDA      2,K1777
000264'132432        SUBZ#    1,2,SZC
000265'000406        JMP      AL150
                 ;   ERROR
000266'040025$       STA      0,BFPT                    ;TYPE "MAX INVALID"
000267'020725        LDA      0,Q15B
000270'006004$       JSR      @P.CR
000271'010026$       ISZ      ERCT
```

```
0005 AL000
000272'020025$          LDA     0,BFPT
                     ;
                     ;GET & CHECK CR
000273'006006$AL150:    JSR     @CK.CR
000274'000401           JMP     .+1
                     ;CHECK MIN < MAX
000275'020461           LDA     0,IFDEF            ;CAN'T CHECK IF ONE LEFT INTACT
000276'101004           MOV     0,0,SZR
000277'000410           JMP     AL160
                     ;
000300'020703           LDA     0,MIN
000301'024703           LDA     1,MAX
000302'106432           SUBZ#   0,1,SZC
000303'000404           JMP     AL160              ;MIN <= MAX
                     ;    ERROR
000304'020711           LDA     0,Q16B
000305'006004$          JSR     @P.CR
000306'010026$          ISZ     ERCT
                     ;CHECK FOR ANY ERRORS IN LINE
000307'024025$AL160:    LDA     1,ERCT
000310'125004           MOV     1,1,SZR
000311'000706           JMP     AL100
                     ;LOOP TO MODIFY TABLE
000312'020022$          LDA     0,LADR1
                     ;    GET TABLE ADDRESSES
000313'034016$AL170:    LDA     3,TAB4A
000314'117000           ADD     0,3
000315'030015$          LDA     2,TAB3A
000316'113000           ADD     0,2
                     ;    LOOP
000317'024023$          LDA     1,LADR2
000320'106433           SUBZ#   0,1,SNC
000321'000676           JMP     AL100              ;END OF LOOP
                     ;
000322'024434           LDA     1,IFDEF
000323'125004           MOV     1,1,SZR
000324'000406           JMP     AL180
                     ;    BOTH SPECIFIED
000325'024656           LDA     1,MIN
000326'045000           STA     1,0,2
000327'024655           LDA     1,MAX
000330'045400           STA     1,0,3
000331'000421           JMP     AL190
                     ;    ONE LEFT INTACT
000332'040017$AL180:    STA     0,ADR              ;SAVE FOR R.PS
000333'020650           LDA     0,MIN
000334'101112           MOVL#   0,0,SZC
000335'021000           LDA     0,0,2              ;GET EXISTING MIN
                     ;
000336'024646           LDA     1,MAX
000337'125112           MOVL#   1,1,SZC
000340'025400           LDA     1,0,3              ;GET EXISTING MAX
                     ;
000341'106432           SUBZ#   0,1,SZC
000342'000405           JMP     AL182              ;OK
                     ;    MIN > MAX
000343'020652           LDA     0,Q16B
000344'006004$          JSR     @P.CR
000345'006013$          JSR     @R.PS
000346'000403           JMP     AL185
```

```
0006 AL000
                    ;    SET MIN & MAX
000347'041000 AL182: STA     0,0,2
000350'045400        STA     1,0,3
000351'020022$AL185: LDA     0,LADR1
                    ;    LOOP
000352'024024$AL190: LDA     1,LINCR
000353'123000        ADD     1,0
000354'040022$       STA     0,LADR1
000355'000736        JMP     AL170
000356'000000 IFDEF: .-.
                     .TITL   AL900   ;END
                    ;
                    ;    END OF ALARM LIMIT SELECTION
                    ;
000357'020610 AL900: LDA     0,AL0A0
000360'024610        LDA     1,AL0A1
000361'030610        LDA     2,AL0A2
000362'002610        JMP     @AL0A3
                    ;
                     .TITL   AL000
                     .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
   0007 AL000

  ADR    000017$ XD      1/09      5/45
  AL000  000104' EN      1/01      2/24      2/25      2/29      5/20
  AL001  000000' EN      1/14      1/15
  AL002  000175' EN      3/40      3/41      3/43
  AL010  000116'         2/42      2/60      3/09
  AL015  000135'         2/39      3/02
  AL020  000141'         2/52      3/07
  AL040  000152'         3/18      3/23
  AL080  000160'         2/56      3/25
  AL090  000164'         3/05      3/30
  AL093  000216'         2/46      4/02      4/10
  AL0A0  000167'         2/29      3/33      6/15
  AL0A1  000170'         2/30      3/34      6/16
  AL0A2  000171'         2/31      3/35      6/17
  AL0A3  000172'         2/32      3/36      6/18
  AL100  000217' EN      3/32      4/03      4/04      4/06      4/13      5/22      5/33
  AL120  000237'         4/20      4/27
  AL130  000247'         4/25      4/29      4/38
  AL140  000263'         4/46      4/53
  AL150  000273'         4/51      4/55      5/04
  AL160  000307'         5/09      5/14      5/20
  AL170  000313'         5/26      6/09
  AL180  000332'         5/37      5/45
  AL182  000347'         5/55      6/02
  AL185  000351'         5/60      6/04
  AL190  000352'         5/43      6/06
  AL900  000357'         4/02      6/11      6/15
  BFPT   000025$ XD      1/09      4/31      4/35      4/57      5/01
  CK.CR  000006$ XD      1/06      5/04
  DIG.   000001$ XD      1/06      4/17      4/43
  ERCT   000026$ XD      1/09      4/34      4/60      5/18      5/20
  FM.ER  000005$ XD      1/06      4/41
  GT.CH  000002$ XD      1/06      2/48      4/38
  G.ADR  000012$ XD      1/07      4/12
  HIADR  000020$ XD      1/08      3/11
  IBUF   000007$ XD      1/07      2/47
  IFDEF  000356'         4/15      4/23      4/49      5/07      5/35      6/10
  IFMOD  000021$ XD      1/09      3/02      3/26
  K1777  000202'         3/48      4/27      4/53
  LADR1  000022$ XD      1/09      5/24      6/04      6/08
  LADR2  000023$ XD      1/09      5/31
  LINCR  000024$ XD      1/09      3/12      3/22      6/06
  MAX    000204'         3/50      4/44      4/50      5/12      5/41      5/50
  MIN    000203'         3/49      4/18      4/24      5/11      5/39      5/46
  MVINI  000010$ XD      1/07      2/37
  P.CR   000004$ XD      1/06      2/35      2/43      3/30      4/07      4/33      4/59
                         5/17      5/58
  Q00    000000'         1/16      3/52
  Q00B   000205'         2/34      3/52
  Q01B   000206'         2/42      3/53
  Q01.2  000206' XN      1/04      3/53
  Q02    000016'         1/30      3/54
  Q02B   000207'         3/07      3/54
  Q06    000041'         1/49      3/55
  Q06B   000210'         3/29      3/55
  Q07B   000211'         3/04      3/56
  Q07.2  000211' XN      1/04      3/56
  Q08B   000212'         3/57      4/06
  Q08.2  000212' XN      1/04      3/57
  0008 AL000

Q14    000060'          2/04     3/58
Q14B   000213'          3/58     4/32
Q15    000067'          2/11     3/59
Q15B   000214'          3/59     4/58
Q16    000076'          2/18     3/60
Q16B   000215'          3/60     5/16     5/57
QRC    000200'          2/59     3/46
QRCM   000201'          3/47     4/39
QRI    000176'          2/50     3/44
QRM    000177'          2/54     3/45
R.LN   000003$ XD       1/06     2/45     4/09
R.PS   000013$ XD       1/07     5/59
TAB1A  000014$ XD       1/08
TAB3A  000015$ XD       1/08     3/13     5/28
TAB4A  000016$ XD       1/08     3/15     5/26
VB15B  000173'          2/41     3/37
VB240  000175' XN       1/03     3/39
VB25B  000174'          3/28     3/38
VB300  000173' XN       1/03     3/37
VB320  000174' XN       1/03     3/38
VB35B  000175'          3/03     3/39
WRN.A  000011$ XD       1/07     3/08
```

```
0001 LG000 MACRO REV 06.50                       19:59:00 10/31/80
                              .TITL   LG000   ;LOGGING
                              .EXTN   VB400,Q13.2
                              .EXTD   P.NL,P.CR,R.LN,DIG.,BN.AS,GT.CH,MUL.U
                              .EXTD   WRN.A,IBUF
          000001              .TXTM   1

                              .ZREL
                              .ENT    LGMNC,LGMNS,LGMNB
   000000-000002 LGMNC:  2            ;MINUTE COUNTER
   000001-000000 LGMNS:  .-.          ;INTERVAL IN MINUTES (LOG CONS)
   000002-000000 LGMNB:  .-.          ;INTERVAL IN MINUTES (LOG B CONS)
                              .NREL
                              .ENT    LG001
                        LG001:
   000000'005040 Q00:    .TXT     /<12> * * * LOGGING INTERVAL * * *<15>/
          025040
          025040
          025040
          046117
          043507
          044516
          043440
          044516
          052105
          051126
          040514
          020052
          020052
          020052
          006400
   000020'044516 Q01:    .TXT     /INTERVAL IS /
          052105
          051126
          040514
          020111
          051440
          000000
   000027'044117 Q02:    .TXT     /HOURS.<15>/
          052522
          051456
          006400
   000033'046511 Q03:    .TXT     /MINUTES.<15>/
          047125
          052105
          051456
          006400
   000040'047105 Q04:    .TXT     /NEW INTERVAL: /
          053440
          044516
          052105
          051126
          040514
          035040
          000000
   000050'044117 Q05:    .TXT     /HOURS OR MINUTES: /
          052522
          051440
          047522
          020115
          044516
```

```
 0002 LG000
01          052524
02          042523
03          035040
04          000000
05 000062'047117 Q06:    .TXT    /NO LOGGING.<15>/
06          020114
07          047507
08          043511
09          047107
10          027015
11          000000
12 000071'040514 Q07:    .TXT    /ALL VALUES IN VOLTS<15>/
13          046040
14          053101
15          046125
16          042523
17          020111
18          047040
19          053117
20          046124
21          051415
22          000000
23 000104'053101 Q12:    .TXT    /VALUE ERROR<15>/
24          046125
25          042440
26          042522
27          051117
28          051015
29          000000
30 000113'042101 Q14:    .TXT    /DANGEROUS OPTION: FAILED SENSORS ARE
31          047107
32          042522
33          047525
34          051440
35          047520
36          052111
37          047516
38          035040
39          043101
40          044514
41          042504
42          020123
43          042516
44          051517
45          051123
46          020101
47          051105
48 000135'020116  NOT DETECTED.<15>/
49          047524
50          020104
51          042524
52          042503
53          052105
54          042056
55          005400
56 000145'000000 Q15:    .TXT    /<0>/
 7
58 000146'000000"Q00B:   Q00*2
59 000147'000040"Q01B:   Q01*2
60 000150'000056"Q02B:   Q02*2
```

```
0003 LG000
000151'000066"003B:   003*2
000152'000100"004B:   004*2
000153'000120"005B:   005*2
000154'000144"006B:   006*2
                      .TITL    LG000    ;SELECTION
                      .ENT     LG000
                ;
                ;     LOGGING INTERVAL SELECTION
000155'040510 LG000:  STA      0,LG0A0              ;SAVE ACCUMULATORS
000156'044510         STA      1,LG0A1
000157'050510         STA      2,LG0A2
000160'054510         STA      3,LG0A3
                ;
000161'020765         LDA      0,000B               ;TYPE "LOGGING"
000162'006002$        JSR      @P.CR
000163'006517         JSR      @VBSUB               ;PROMPTING
                ;
000164'004521 LG010:  JSR      LG100                ;TYPE PRESENT INTERVAL
                ;
000165'020765 LG015:  LDA      0,004B               ;TYPE "ENTER"
000166'006002$        JSR      @P.CR
000167'006003$        JSR      @R.LN                ;READ RESPONSE
000170'000471         JMP      LG090                ;JUST CR:  RETURN
                ;
000171'020011$        LDA      0,IBUF               ;DECODE NUMBER
000172'006004$        JSR      @DIG.
000173'040504         STA      0,LG091              ;SAVE BYTE POINTER
                ;CHECK FOR 1 OR 2 DIGITS
000174'034504         LDA      3,K2                 ;CONSTANT 2
000175'151004         MOV      2,2,SZR
000176'156433         SUBZ#    2,3,SNC
000177'000543         JMP      LG210                ;FORMAT ERROR
                ;SAVE VALUE
000200'044501         STA      1,LG093
000201'125005         MOV      1,1,SNR
000202'000446         JMP      LG070                ;CANCEL LOGGING
                ;CHECK NEXT CHAR FOR BLANK
000203'020474         LDA      0,LG091
000204'006006$        JSR      @GT.CH               ;GET NEXT CHAR
000205'030467         LDA      2,QRBL
000206'132404         SUB      1,2,SZR
000207'000402         JMP      LG030                ;NOT BLANK, CHECK H OR M
                ;CHECK NEXT CHAR FOR H OR M
000210'006006$LG020:  JSR      @GT.CH               ;GET NEXT CHAR
                ;     CHECK FOR H
000211'030464 LG030:  LDA      2,QRH
000212'132405         SUB      1,2,SNR
000213'000412         JMP      LG040                ;PROCESS HOURS
                ;     CHECK FOR M
000214'030462         LDA      2,QRM
000215'132405         SUB      1,2,SNR
000216'000415         JMP      LG050                ;PROCESS MINUTES
                ;     ERROR - NOT HOURS OR MINUTES
000217'020734         LDA      0,005B               ;TYPE "HOURS OR MINS?"
000220'006002$        JSR      @P.CR
000221'006003$        JSR      @R.LN                ;READ RESPONSE
000222'000742         JMP      LG010                ;JUST CR: TRY AGAIN
000223'020011$        LDA      0,IBUF
000224'000764         JMP      LG020
                ;HOURS
```

```
                      .ENT    LG040
000225'024454 LG040:  LDA     1,LG093
000226'044444         STA     1,LGHR
000227'030444         LDA     2,D60
000230'006007$        JSR     @MUL.U               ;TIMES 60 FOR
000231'044001-        STA     1,LGMNS              ;INTERVAL IN MINUTES
000232'000413         JMP     LG060
              ;MINUTES
000233'024446 LG050:  LDA     1,LG093
              ;
000234'030437         LDA     2,D60
000235'146432         SUBZ#   2,1,SZC
000236'000507         JMP     LG220                ;VALUE ERROR
000237'030432         LDA     2,LGMIN
000240'146433         SUBZ#   2,1,SNC
000241'000504         JMP     LG220                ;VALUE ERROR
              ;
000242'044001-        STA     1,LGMNS              ;INTERVAL IN MINUTES
000243'126400         SUB     1,1
000244'044426         STA     1,LGHR
              ;COMMON PROCESSING - INITIALIZE LOGGING
000245'024433 LG060:  LDA     1,K2
000246'044000-        STA     1,LGMNC              ;MINUTE COUNT = 2
000247'000411         JMP     LG080
              ;NO LOGGING
000250'044422 LG070:  STA     1,LGHR
000251'044001-        STA     1,LGMNS
000252'044000-        STA     1,LGMNC
              ;    WARNING
000253'020430         LDA     0,Q14B               ;PRINT WARNING
000254'006002$        JSR     @P.CR
000255'020427         LDA     0,Q15B               ;CHANCE TO ABORT
000256'006010$        JSR     @WRN.A
000257'000706         JMP     LG015
              ;RETURN FROM THIS ROUTINE
000260'004425 LG080:  JSR     LG100                ;TYPE PRESENT INTERVAL
000261'020404 LG090:  LDA     0,LG0A0
000262'024404         LDA     1,LG0A1
000263'030404         LDA     2,LG0A2
000264'002404         JMP     @LG0A3
              ;
000265'000000 LG0A0:  .-.
000266'000000 LG0A1:  .-.
000267'000000 LG0A2:  .-.
000270'000000 LG0A3:  .-.
                      .ENT    LGMIN
000271'000002 LGMIN:  2.      ;MINIMUM INTERVAL
000272'000001 LGHR:   1
000273'000074 D60:    60.
000274'000040 QRBL:   40
000275'000110 QRH:    "H
000276'000115 QRM:    "M
000277'000000 LG091:  .-.     ;BUFFER BYTE POINTER
000300'000002 K2:     2
000301'000000 LG093:  .-.     ;TEMP STORAGE
000302'177777 VBSUB:  VB400
000303'000226"Q14B:   Q14*2
000304'000312"Q15B:   Q15*2
                      .TITL   LG100   ;PRESENT STATUS
                      .ENT    LG100
```

```
                  ;
                  ;ROUTINE TO TYPE PRESENT INTERVAL
                  ;
000305'054426 LG100: STA     3,LG1A3
000306'024754        LDA     1,LGHR          ;CHECK HOURS
000307'125004        MOV     1,1,SZR
000310'000407        JMP     LG110
000311'024001-       LDA     1,LGMNS         ;CHECK MINUTES
000312'125004        MOV     1,1,SZR
000313'000404        JMP     LG110
                  ;
000314'020640        LDA     0,Q06B          ;PRINT "NO LG"
000315'006002$       JSR     @P.CR
000316'000414        JMP     LG190
                  ;CONVERT TO ASCII
000317'020421 LG110: LDA     0,Q08B          ;BYTE POINTER
000320'006005$       JSR     @BN.AS          ;CONVERT
                  ;PRINT
000321'020626        LDA     0,Q01B          ;TYPE "PRESENT INTERVAL"
000322'006002$       JSR     @P.CR
000323'020416        LDA     0,Q08B4         ;TYPE 2 DIGITS
000324'006001$       JSR     @P.NL
000325'020623        LDA     0,Q02B          ;HOURS
000326'024744        LDA     1,LGHR
000327'125005        MOV     1,1,SNR
000330'020621        LDA     0,Q03B          ;MINUTES
000331'006001$       JSR     @P.NL           ;TYPE UNITS
                  ;RETURN
000332'002401 LG190: JMP     @LG1A3
                  ;STORAGE
000333'000000 LG1A3: .-.                     ;RETURN ADDRESS
000334'054130 Q08:   .TXT    /XXXXXX /
       054130
       054130
       020000
000340'000570"Q08B:  Q08*2
000341'000674"Q08B4: Q08*2+4
                  ;ERROR ROUTINES
000342'020406 LG210: LDA     0,Q13B          ;TYPE "FORMAT ERROR"
000343'006002$       JSR     @P.CR
000344'000621        JMP     LG015
                  ;
000345'020404 LG220: LDA     0,Q12B          ;TYPE "VALUE ERROR"
000346'006002$       JSR     @P.CR
000347'000615        JMP     LG015
000350'177777 Q13B:  Q13.2
000351'000210"Q12B:  Q12*2
                     .TITL   LG500   ;LOGGING TASK
                     .ENT    LG500
                     .EXTN   .KILL,.REC
                     .EXTD   HIADR,TAB1A,TLGA,LGSAV,P.VAL,P.DT
                     .EXTD   LINCR,ADR,IFLOG,IFVLT
                  ;    LOGGING TASK
                  ;
000352'102520 LG500: SUBZL   0,0
000353'040022$       STA     0,IFLOG
000354'020001-       LDA     0,LGMNS         ;CHECK IF LOGGING REQUESTED
000355'101004        MOV     0,0,SZR
000356'000404        JMP     LG510
```

```
0006 LI300
000357'020002-          LDA     0,LGMNB          ;CHECK LOGGING B
000360'101005           MOV     0,0,SNR
000361'177777           .KILL                    ;IF NOT, KILL
                ;WAIT FOR DATA TO BE READY
000362'020014$LG510:    LDA     0,TLGA
000363'040015$          STA     0,LGSAV
000364'034460           LDA     3,LGMSA
000365'102400           SUB     0,0              ;WIPE OUT ANY
000366'041400           STA     0,0,3            ;MESSAGE WAITING
000367'161000           MOV     3,0              ;WAIT FOR NEXT
000370'177777           .REC
000371'102400           SUB     0,0
000372'040015$          STA     0,LGSAV          ;SUPPRESS LOGGING SAVE
                ;PRINT HEADER
000373'020453           LDA     0,QLCR
000374'006017$          JSR     @P.DT            ;PRINT DATE & TIME
000375'020023$          LDA     0,IFVLT
000376'101005           MOV     0,0,SNR
000377'000403           JMP     LG515
000400'020447           LDA     0,Q07B
000401'006002$          JSR     @P.CR            ;PRINT "ALL IN VOLTS"
                ;LOOP TO PRINT ALL VALUES
000402'020437 LG515:    LDA     0,LOADR          ;LOW ADR VARIABLE
000403'040021$          STA     0,ADR
000404'024012$          LDA     1,HIADR
000405'106400           SUB     0,1
000406'044020$          STA     1,LINCR
                ;LOOP TO PRINT LINE
000407'024021$LG520:    LDA     1,ADR
                ;CHECK RATE
000410'034013$          LDA     3,TAB1A
000411'137000           ADD     1,3
000412'021400           LDA     0,0,3
000413'030432           LDA     2,MKRAT
000414'113405           AND     0,2,SNR
000415'000420           JMP     LG540            ;LOOP INCR
                ;CHECK LOGGING "B"
000416'030002-          LDA     2,LGMNB
000417'151005           MOV     2,2,SNR
000420'000404           JMP     LG530            ;LOGGING A, GO PRINT
                ;
000421'030421           LDA     2,MKLGB          ;LOGGING B, CHECK IF SELECTED
000422'113405           AND     0,2,SNR
000423'000412           JMP     LG540
                ;CONVERT & PRINT ADR
000424'020424 LG530:    LDA     0,Q50B
000425'006005$          JSR     @BN.AS
000426'020423           LDA     0,Q50B3
000427'006002$          JSR     @P.CR
                ;GET VALUE
000430'030021$          LDA     2,ADR
000431'034014$          LDA     3,TLGA
000432'157000           ADD     2,3
000433'025400           LDA     1,0,3            ;GET VALUE
000434'006016$          JSR     @P.VAL
                ;INCR LOOP
000435'010021$LG540:    ISZ     ADR
000436'014020$          DSZ     LINCR
000437'000750           JMP     LG520
                ;***PRINT CALCS
```

```
 0007 LG000
 000440'000722          JMP     LG510
                  ;
                        .ENT    LOADR,LGMSG
 000441'000001 LOADR:   1       ;CHANGE FOR TESTING
 000442'020000 MKLGB:   1B2
 000443'000000 LGMSG:   .-.
 000444'000443'LGMSA:   LGMSG
 000445'016000 MKRAT:   7B5
 000446'006400 QLCR:    6400
 000447'000162"Q07B:    Q07*2
 000450'001124"Q50B:    Q50*2
 000451'001127"Q50B3:   Q50*2+3
                  ;
 000452'054130 Q50:     .TXT    /XXXADR /
        054101
        042122
        020000
                        .TITL   LG000
                        .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
  0008 LG000

ADR    000021$ XD      5/52      5/24      5/29      6/51      6/57
BN.AS  000005$ XD      1/03      5/17      5/47
DS0    000273'         4/04      4/11      4/49
DIG.   000004$ XD      1/03      3/26
GT.CH  000006$ XD      1/03      3/39      3/44
HIADR  000012$ XD      5/51      5/25
IBUF   000011$ XD      1/04      3/25      3/58
IFLOG  000022$ XD      5/52      5/56
IFVLT  000023$ XD      5/52      6/17
K2     000300'         3/29      4/22      4/54
LG000  000155' EN      1/01      3/05      3/06      3/09      7/18
LG001  000000' EN      1/13      1/14
LG010  000164'         3/18      3/57
LG015  000165'         3/20      4/34      5/41      5/45
LG020  000210'         3/44      3/59
LG030  000211'         3/42      3/46
LG040  000225' EN      3/48      4/01      4/02
LG050  000233'         3/52      4/09
LG060  000245'         4/07      4/22
LG070  000250'         3/36      4/26
LG080  000260'         4/24      4/36
LG090  000261'         3/23      4/37
LG091  000277'         3/27      3/38      4/53
LG093  000301'         3/34      4/02      4/09      4/55
LG0A0  000265'         3/09      4/37      4/42
LG0A1  000266'         3/10      4/38      4/43
LG0A2  000267'         3/11      4/39      4/44
LG0A3  000270'         3/12      4/40      4/45
LG100  000305' EN      3/18      4/36      4/59      4/60      5/04
LG110  000317'         5/07      5/10      5/16
LG190  000332'         5/14      5/29
LG1A3  000333'         5/04      5/29      5/31
LG210  000342'         3/32      5/39
LG220  000345'         4/13      4/16      5/43
LG500  000352' EN      5/48      5/49      5/55
LG510  000362'         5/59      6/05      7/01
LG515  000402'         6/19      6/23
LG520  000407'         6/29      6/59
LG530  000424'         6/40      6/46
LG540  000435'         6/36      6/44      6/57
LGHR   000272'         4/03      4/20      4/26      4/48      5/05      5/24
LGMIN  000271' EN      4/14      4/46      4/47
LGMNB  000002- EN      1/08      1/11      6/01      6/38
LGMNC  000000- EN      1/08      1/09      4/23      4/28
LGMNS  000001- EN      1/08      1/10      4/06      4/18      4/27      5/08      5/57
LGMSA  000444'         6/07      7/07
LGMSC  000443' EN      7/03      7/06      7/07
LGSAV  000015$ XD      5/51      6/06      6/13
LINCR  000020$ XD      5/52      6/27      6/58
LOADR  000441' EN      6/23      7/03      7/04
MKLGB  000442'         6/42      7/05
MKRAT  000445'         6/34      7/08
MUL.U  000007$ XD      1/03      4/05
P.CR   000002$ XD      1/03      3/15      3/21      3/55      4/31      5/13      5/20
                       5/40      5/44      6/21      6/49
P.DT   000017$ XD      5/51      6/16
P.NL   000001$ XD      1/03      5/22      5/27
P.VAL  000016$ XD      5/51      6/55
000    000000'         1/15      2/58
```

```
  0003 LG300

Q00B   000146'          2/58     3/14
Q01    000020'          1/31     2/59
Q01B   000147'          2/59     5/19
Q02    000027'          1/38     2/60
Q02B   000150'          2/60     5/23
Q03    000033'          1/42     3/01
Q03B   000151'          3/01     5/26
Q04    000040'          1/47     3/02
Q04B   000152'          3/02     3/20
Q05    000050'          1/55     3/03
Q05B   000153'          3/03     3/54
Q06    000062'          2/05     3/04
Q06B   000154'          3/04     5/12
Q07    000071'          2/12     7/10
Q07B   000447'          6/20     7/10
Q08    000334'          5/32     5/36     5/37
Q08B   000340'          5/16     5/36
Q08B4  000341'          5/21     5/37
Q12    000104'          2/23     5/47
Q12B   000351'          5/43     5/47
Q13B   000350'          5/39     5/46
Q13.2  000350' XN       1/02     5/46
Q14    000113'          2/30     4/57
Q14B   000303'          4/30     4/57
Q15    000145'          2/56     4/58
Q15B   000304'          4/32     4/58
Q50    000452'          7/11     7/12     7/14
Q50B   000450'          6/46     7/11
Q50B3  000451'          6/48     7/12
QLCR   000446'          6/15     7/09
QRBL   000274'          3/40     4/50
QRH    000275'          3/46     4/51
QRM    000276'          3/50     4/52
R.LN   000003$ XD       1/03     3/22     3/56
TAB1A  000013$ XD       5/51     6/31
TLGA   000014$ XD       5/51     6/05     6/52
VB400  000302' XN       1/02     4/56
VBSUB  000302'          3/16     4/56
WRN.A  000010$ XD       1/04     4/33
.KILL  000361' XN       5/50     6/03
.REC   000370' XN       5/50     6/11
```

```
0001 MN000 MACRO REV 06.50                       20:11:40 10/31/80
                                  .TITL   MN000   .MONITORING
                         ;
                                  .EXTN   VB500.VB600.Q12.2.Q13.2
                                  .EXTD   P.NL,P.CR,R.LN,DIG.,BN.AS,CK.CR
                                  .EXTD   IBUF,HIADR,TAB1A
                         ;
             000001               .TXTM   1
                         ;
                                  .ZREL
                                  .ENT    MNADR,MNBAD,MNVAL
     000000-000000 MNADR: 0
     000001-000000 MNBAD: 0
     000002-000000 MNVAL: .-.
                         ;
                                  .NREL
                                  .ENT    MN001
                   MN001:
     000000'005040 Q00:   .TXT    /<12> * * * MONITORING * * *<15>/
            025040
            025040
            025040
            046517
            047111
            052117
            051111
            047107
            020052
            020052
            020052
            006400
     000015'047117 Q01:   .TXT    /NO MONITORING.<15>/
            020115
            047516
            044524
            047522
            044516
            043456
            006400
     000025'046517 Q02:   .TXT    /MONITORING ADR /
            047111
            052117
            051111
            047107
            020101
            042122
            020000
     000035'040514 Q03:   .TXT    /ALL VALUES IN /
            046040
            053101
            046125
            042523
            020111
            047040
            000000
     000045'040504 Q04:   .TXT    /ADR: /
            051072
            020000
     000050'054130 Q06:   .TXT    /XXXXXX.<15>/
            054130
            054130
```

```
0002 MN000
              027015
              000000
000055'040504 Q14:    .TXT    /ADR NOT SAMPLED<15>/
              051040
              047117
              052040
              051501
              046520
              046105
              042015
              000000
000056'047117 Q21:    .TXT    /NO ALARM B.<15>/
              020101
              046101
              051115
              020102
              027015
              000000
000075'040514 Q22:    .TXT    /ALARM B ON ADR /
              040522
              046440
              041040
              047516
              020101
              042122
              020000
000105'040514 Q24:    .TXT    /ALARM B ADR: /
              040522
              046440
              041040
              040504
              051072
              020000
                      .TITL   MN000   ;MONITORING SELECTION
                      .ENT    MN000
                     ;
                     ;   MONITORING SELECTION
                     ;
000114'040416 MN000:  STA     0,MN0A0
000115'044415         STA     1,MN0A1
000116'050416         STA     2,MN0A2
000117'054416         STA     3,MN0A3
                     ;
000120'020416         LDA     0,Q00B          ;TYPE "MONITORING"
000121'006002$        JSR     @P.CR
                     ;
000122'030415         LDA     2,ADMON         ;DO MONITORING
000123'004430         JSR     MN400
                     ;
000124'030414         LDA     2,ADALM         ;DO ALARM "B"
000125'004426         JSR     MN400
                     ;
                     ;        END OF ROUTINE
                     ;
000126'020404         LDA     0,MN0A0
000127'024404         LDA     1,MN0A1
000130'030404         LDA     2,MN0A2
000131'002404         JMP     @MN0A3
                     ;
000132'000000 MN0A0:  .-.
```

```
0003 MN000
000133'000000 MN0A1:  .-.
000134'000000 MN0A2:  .-.
000135'000000 MN0A3:  .-.
000136'000000"Q00S:   Q00*2
000137'000141'ADMON:  VB5
000140'000146'ADALM:  VB6
                   ;
000141'177777 VB5:    VB500
000142'000032"        Q01*2
000143'000052"        Q02*2
000144'000112"        Q04*2
000145'000000-        MNADR
                   ;
000146'177777 VB6:    VB600
000147'000154"        Q21*2
000150'000172"        Q22*2
000151'000212"        Q24*2
000152'000001-        MNBAD
                      .TITL   MN400   ;MONITOR OR ALARM B
                      .ENT    MN400

                   ;
                   ;  ROUTINE TO DO I/O FOR EITHER
                   ;
000153'054450 MN400:  STA     3,MN4A3
000154'050446         STA     2,MN4A2               ;PROMPTING
000155'007000         JSR     @0,2
                   ;                                ;TYPE PRESENT STATUS
000156'004452         JSR     MN100
                   ;                                ;WHICH PHASE
000157'030443 MN410:  LDA     2,MN4A2               ;TYPE "ENTER"
000160'021003         LDA     0,3,2
000161'006002$        JSR     @P.CR                 ;READ RESPONSE
000162'006003$        JSR     @R.LN                 ;JUST CR, RETURN
000163'000436         JMP     MN460
                   ;                                ;DECODE NUMBER
000164'020007$        LDA     0,IBUF
000165'006004$        JSR     @DIG.                 ;CHECK DIGITS
000166'151004         MOV     2,2,SZR               ;ADR FOUND
000167'000404         JMP     MN420
                   ;    ERROR                       ;FORMAT ERROR
000170'020435         LDA     0,Q13S
000171'006002$MNERR:  JSR     @P.CR
000172'000765         JMP     MN410
                   ;ADDRESS                         ;CHECK IF ZERO
000173'125005 MN420:  MOV     1,1,SNR
000174'000416         JMP     MN440
                   ;
000175'034010$        LDA     3,HIADR
000176'166433         SUBZ#   3,1,SNC               ;ERROR - OUT OF RANGE ADR
000177'000403         JMP     MN430                 ;TYPE "ADR OUT OF RANGE"
000200'020424         LDA     0,Q12S
000201'000770         JMP     MNERR
                   ;CHECK IF ADR IN SAMPLING PATTERN
000202'034011$MN430:  LDA     3,TAB1A
000203'137000         ADD     1,3                   ;GET WORD
000204'035400         LDA     3,0,3
000205'030422         LDA     2,MKRAT               ;ISOLATE RATE
000206'173404         AND     3,2,SZR               ;OK
000207'000403         JMP     MN440                 ;TYPE "ADR NOT SAMPLED"
000210'020416         LDA     0,Q14S
```

```
000211'000760           JMP      MNERR
                      ;   GET & CHECK CR
000212'044016$MN440:    STA      1,ADR
000213'006006$          JSR      @CK.CR
000214'000743           JMP      MN410
                      ;   SAVE ADR
000215'024016$MN450:    LDA      1,ADR
000216'030404           LDA      2,MN4A2
000217'047004           STA      1,@4,2
000220'004410           JSR      MN100              ;TYPE PRESENT SETTING
000221'002402 MN460:    JMP      @MN4A3             ;RETURN
              ;
000222'000000 MN4A2:    .-.
000223'000000 MN4A3:    .-.
000224'177777 Q12B:   Q12.2
000225'177777 Q13B:   Q13.2
000226'000132"Q14B:   Q14*2
000227'015000 MKRAT:  7B5
                        .TITL    MN100   ;PRESENT STATUS
                        .ENT     MN100
              ;
              ;   ROUTINE TO TYPE PRESENT STATUS
              ;
000230'054417 MN100:    STA      3,MN1A3
000231'030771           LDA      2,MN4A2            ;WHICH PHASE
              ;MONITORING SENSOR
000232'027004           LDA      1,@4,2             ;CHECK ADR
000233'125005           MOV      1,1,SNR
000234'000410           JMP      MN130              .NO SENSOR
              ;   CONVERT TO ASCII
000235'020413           LDA      0,Q06B
000236'006005$          JSR      @BN.AS
              ;   PRINT
000237'021002           LDA      0,2,2              ;PRINT "ADR"
000240'006002$          JSR      @P.CR
000241'020410           LDA      0,Q06B3            ;PRINT ADR
000242'006001$          JSR      @P.NL
000243'000403           JMP      MN190
              ;NO MONITORING
000244'021001 MN130:    LDA      0,1,2
000245'006002$          JSR      @P.CR              ;TYPE "NO MONITORING"
              ;
000246'002401 MN190:    JMP      @MN1A3
              ;STORAGE
000247'000000 MN1A3:    .-.
000250'000120"Q06B:   Q06*2
000251'000123"Q06B3:  Q06*2+3
                        .TITL    MN500   ;PRINTING TASK
                        .ENT     MN500
                        .EXTD    P.DA,P.TM,P.NL,P.VAL,R.PS,ADR,IFVLT,IFLOG
                        .EXTN    LG500,.REC,STAB
              ;
              ;   MONITORING PRINTING TASK
              ;CHECK FOR MONITORING ADR
000252'020000-MN500:    LDA      0,MNADR
000253'040016$          STA      0,ADR
000254'101005           MOV      0,0,SNR
000255'002440           JMP      @TKLOG
              ;
              ;   MONITOR ADR
```

```
 0005 MN000
01                        ;
02 000256'102400          SUB     0,0
03 000257'040020$         STA     0,IFLOG
04 000260'020432          LDA     0,050B                ;SKIP A LINE
05 000261'006002$         JSR     @P.CR
06                        ;PRINT FIRST LINE
07 000262'102400          SUB     0,0
08 000263'006012$         JSR     @P.DA                 ;PRINT DATE W/ NO CR
09 000264'020427          LDA     0,Q513
10 000265'006001$         JSR     @P.NL                 ;PRINT "MONITORING ADR"
11 000266'006015$         JSR     @R.PS                 ;PRINT ADR & STATUS
12                        ;PRINT SECOND LINE
13 000267'020422          LDA     0,Q03B
14 000270'006002$         JSR     @P.CR                 ;PRINT "ALL VALUES IN"
15 000271'034425          LDA     3,ASTAB               ;TABLE ADDR
16 000272'030016$         LDA     2,ADR                 ;ADR FOR INDEXING
17 000273'020017$         LDA     0,IFVLT               ;CHECK IF ALL IN VOLTS
18 000274'101005          MOV     0,0,SNR
19 000275'157000          ADD     2,3
20 000276'023400          LDA     0,@0,3                ;GET BYTE POINTER
21 000277'006001$         JSR     @P.NL                 ;PRINT UNITS
22                        ;SKIP A LINE
23 000300'020412          LDA     0,050B
24 000301'006002$         JSR     @P.CR
25                        ;WAIT FOR VALUE
26 000302'020412 MN530:   LDA     0,MNMSA
27 000303'177777          .REC
28                        ;PRINT TIME W/ NO CR
29 000304'102400          SUB     0,0
30 000305'006013$         JSR     @P.TM
31                        ;CONVERT & PRINT VALUE
32 000306'030000-         LDA     2,MNADR
33 000307'006014$         JSR     @P.VAL
34 000310'000772          JMP     MN530
35                        ;STORAGE
36 000311'000072"Q03B:    Q03*2
37 000312'000127"Q50B:    Q06*2+7 ;CR
38 000313'000052"Q513:    Q02*2
39 000314'000002-MNMSA:   MNVAL
40 000315'177777 TKLOG:   LG500
41 000316'177777 ASTAB:   STAB
                          .TITL    MN000
43                        .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
  3005 MN000

  ADALM 000140'          2/50     3/05
  ADMON 000137'          2/47     3/05
  ADR   000316$ XD       4/07     4/07     4/50     4/55     5/16
  ASTAB 000316'          5/15     5/41
  EN.AS 000005$ XD       1/04     4/32
  CK.CR 000006$ XD       1/04     4/04
  DIG.  000004$ XD       1/04     3/37
  HIADR 000010$ XD       1/05     3/40
  IBUF  000007$ XD       1/05     3/36
  IFLOC 000020$ XD       4/50     5/03
  IFVLT 000017$ XD       4/50     5/17
  LC500 000315' XN       4/51     5/40
  MKRAT 000227'          3/57     4/18
  MN000 000114' EN       1/01     2/34     2/35     2/39     5/42
  MN001 000000' EN       1/15     1/17
  MN040 000132'          2/39     2/55     2/60
  MN041 000133'          2/40     2/56     3/01
  MN042 000134'          2/41     2/57     3/02
  MN043 000135'          2/42     2/58     3/03
  MN100 000230' EN       3/28     4/10     4/19     4/20     4/24
  MN130 000244'          4/39     4/40
  MN190 000245'          4/38     4/43
  MN143 000247'          4/24     4/43     4/45
  MN400 000153' EN       2/48     2/51     3/19     3/20     3/24
  MN410 000157'          3/30     3/43     4/05
  MN420 000173'          3/39     3/45
  MN430 000202'          3/50     3/54
  MN440 000212'          3/46     3/59     4/03
  MN450 000215'          4/07
  MN460 000221'          3/34     4/11
  MN442 000222'          3/35     3/30     4/08     4/13     4/25
  MN443 000223'          3/24     4/11     4/14
  MN500 000252' EN       4/48     4/49     4/55
  MN530 000302'          5/25     5/34
  MNADR 000000- EN       1/10     1/11     3/12     4/55     5/32
  MNBAD 000001- EN       1/10     1/12     3/18
  MNERR 000171'          3/42     3/52     4/01
  MNMSA 000314'          5/26     5/39
  MNVAL 000002- EN       1/10     1/13     5/39
  P.CR  000002$ XD       1/04     2/45     3/32     3/42     4/35     4/41     5/05
                         5/14     5/24
  P.DA  000012$ XD       4/50     5/08
  P.NL  000001$ XD       1/04     4/37     4/50     5/10     5/21
  P.TM  000013$ XD       4/50     5/30
  P.VAL 000014$ XD       4/50     5/33
  000   000000'          1/18     3/04
  0008  000136'          2/44     3/04
  001   000015'          1/31     3/09
  002   000025'          1/39     3/10     5/38
  003   000035'          1/47     5/36
  0038  000311'          5/13     5/36
  004   000045'          1/55     3/11
  005   000050'          1/58     4/46     4/47     5/37
  0058  000250'          4/31     4/46
  00583 000251'          4/36     4/47
  0128  000224'          3/51     4/15
  012.2 000224' XN       1/03     4/15
  0138  000225'          3/41     4/16
  013.2 000225' XN       1/03     4/16
```

```
0007 XN000

Q14    000035'          2/03     4/17
Q149   000226'          3/50     4/17
Q21    000066'          3/13     3/15
Q22    000075'          2/19     3/16
Q24    000105'          2/27     3/17
Q533   000712'          5/04     5/23     5/37
Q512   000713'          5/09     5/38
R.LN   000003$ XD       1/04     3/33
R.PB   000015$ XD       4/50     5/11
STAB   000716' XN       4/51     5/41
TAB1A  000011$ XD       1/05     3/54
TKLOG  000315'          4/58     5/40
VB5    000141'          3/05     3/08
VB500  000141' XN       1/03     3/08
VB6    000146'          3/05     3/14
VB600  000146' XN       1/03     3/14
.REC   000303' XN       4/51     5/27
```

```
0001 RM100 MACRO REV 06.50                    08:30:09 10/31/80
                              .TITL   RM000   ;RECORDING MODE
                        ;
                              .EXTN   UB700
                              .EXTD   P.CR,P.NL,R.LN,GT.CH,DIG.,BN.AS,FM.ER
                              .EXTD   IBUF,MTFLG,CK.CR
                        ;
        000001                .TXTM   1
                        ;
                              .ZREL
                              .ENT    RMCNT
   000000-000036 RMCNT:  30.
                              .NREL
                              .ENT    RM001
                        RM001:
   000000'005040 Q00:         .TXT    /<12> * * * RECORDING MODE * * *<15>/
          025040
          025040
          025040
          051105
          041517
          051104
          044516
          043440
          046517
          042105
          020052
          020052
          020052
          006400
   000017'051105 Q01:         .TXT    /RECORDING.<15>/
          041517
          051104
          044516
          043456
          006400
   000025'051524 Q02:         .TXT    /STANDBY, ALARM TRIGGERS RECORDING FOR /
          040516
          042102
          054454
          020101
          046101
          051115
          020124
          051111
          043507
          042522
          051440
          051105
          041517
          051104
          044516
          043440
          043117
          051040
          000000
   000051'051505 Q03:         .TXT    /SELECT RECORDING OR STANDBY: /
          046105
          041524
          020122
          042503
```

```
                047522
                042111
                047107
                020117
                051040
                051524
                040516
                042102
                054472
                020000
   000070'054130 Q05:    .TXT    /XXXXXX /
                054130
                054130
                020000
   000074'051505 Q06:    .TXT    /SECONDS.<15>/
                041517
                047104
                051456
                006400
   000101'047105 Q07:    .TXT    /NEW DURATION: /
                053440
                042125
                051101
                052111
                047516
                035040
                000000
                         ;
   000111'000000"Q00B:   Q00*2
   000112'000036"Q01B:   Q01*2
   000113'000052"Q02B:   Q02*2
   000114'000122"Q03B:   Q03*2
   000115'000202"Q07B:   Q07*2
                         .TITL   RM000   ;SELECT
                         .ENT    RM000
                         ;
                         ;   RECORDING MODE SELECTION
                         ;
   000116'040453 RM000:  STA     0,RM0A0
   000117'044453         STA     1,RM0A1
   000120'050453         STA     2,RM0A2
   000121'054453         STA     3,RM0A3
                         ;
   000122'020767         LDA     0,Q00B              ;TYPE "RECORDING MODE"
   000123'006001$        JSR     @P.CR
   000124'006454         JSR     @VBSUB              ;PROMPTING
                         ;
   000125'004454         JSR     RM100               ;TYPE PRESET STATUS
                         ;
   000126'020766 RM010:  LDA     0,Q03B              ;TYPE "SELECT"
   000127'006001$        JSR     @P.CR
   000130'006003$        JSR     @R.LN               ;READ RESPONSE
   000131'000434         JMP     RM090               ;JUST CR; RETURN
                         ;
   000132'020010$        LDA     0,IBUF
   000133'006004$        JSR     @GT.CH              ;GET CHARACTER
                         ;
   000134'020441         LDA     0,QRR               ;CHECK FOR R
   000135'122405         SUB     1,0,SNR
   000136'000424         JMP     RM040               ;STORE 1
```

```
0003 RM100
                     ;
000137'020437          LDA     0,QRS           ;CHECK FOR S
000140'122404          SUB     1,0,SZR
000141'000765          JMP     RM010           ;TRY AGAIN
                     ;STANDBY MODE - HOW LONG TO RECORD?
000142'020753 RM020:   LDA     0,Q07B          ;ASK FOR NEW
000143'006001$         JSR     @P.CR
000144'006003$         JSR     @R.LN           ;GET RESPONSE
000145'000414          JMP     RM030           ;JUST CR
                     ;   DECODE & CHECK
000146'020010$         LDA     0,IBUF
000147'006005$         JSR     @DIG.
000150'151005          MOV     2,2,SNR
000151'000771          JMP     RM020           ;ZERO DIGITS - ERROR
                     ;
000152'030425          LDA     2,RMLIM
000153'132433          SUBZ#   1,2,SNC
000154'000766          JMP     RM020           ;> LIMIT - ERROR
                     ;   SAVE RESPONSE
000155'125120          MOVZL   1,1             ;CONVERT FROM SECONDS TO TAPE WR
000156'044000-         STA     1,RMCNT
                     ;   GET & CHECK CR
000157'006012$         JSR     @CK.CR
000160'000762          JMP     RM020
                     ;   RECORDING INDICATOR
000161'102401 RM030:   SUB     0,0,SKP         ;GET 0 FOR STANDBY
000162'102000 RM040:   ADC     0,0             ;GET -1 FOR RECORDING
000163'040011$         STA     0,MTFLG         ;STORE 0 OR -1
                     ;
000164'004415          JSR     RM100           ;TYPE NEW STATUS
000165'020404 RM090:   LDA     0,RM0A0
000166'024404          LDA     1,RM0A1
000167'030404          LDA     2,RM0A2
000170'002404          JMP     @RM0A3
                     ;STORAGE
000171'000000 RM0A0:   .-.
000172'000000 RM0A1:   .-.
000173'000000 RM0A2:   .-.
000174'000000 RM0A3:   .-.
000175'000122 QRR:     122
000176'000123 QRS:     123
000177'000074 RMLIM:   60.
000200'177777 VBSUB:   VB700
                       .TITL   RM100   ;PRESENT STATUS
                       .ENT    RM100
                     ;
                     ;   REPORT PRESENT RECORDING MODE
                     ;
000201'054421 RM100:   STA     3,RM1A3
000202'020710          LDA     0,Q01B          ;RECORDING MODE
000203'024011$         LDA     1,MTFLG
000204'125123          MOVZL   1,1,SNC
000205'020706          LDA     0,Q02B          ;IF ZERO, STANDBY
000206'006001$         JSR     @P.CR
                     ;
000207'125122          MOVZL   1,1,SZC
000210'002412          JMP     @RM1A3          ;RECORDING MODE
                     ;   REPORT PRESENT RECORDING DURATION
000211'020412          LDA     0,Q05B
000212'024000-         LDA     1,RMCNT
```

```
0004 RM100
000213'125220        MOVZR   1,1              ;CONVERT FROM TAPE WRITES TO SEC
000214'006006$       JSR     @BN.AS
000215'020407        LDA     0,Q05B4
000216'006002$       JSR     @P.NL
                  ;
000217'020406        LDA     0,Q06B
000220'006002$       JSR     @P.NL
000221'002401        JMP     @RM1A3
                  ;
000222'000000 RM1A3: .-.
000223'000160"Q05B:  Q05*2
000224'000164"Q05B4: Q05*2+4
000225'000170"Q06B:  Q06*2
                     .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
0005 RM100

BN.AS 000006$ XD    1/04    4/02
CK.CR 000012$ XD    1/05    3/23
DIG.  000005$ XD    1/04    3/12
FM.ER 000007$ XD    1/04
GT.CH 000004$ XD    1/04    2/56
IBUF  000010$ XD    1/05    2/55    3/11
MTFLG 000011$ XD    1/05    3/28    3/51
P.CR  000001$ XD    1/04    2/45    2/51    3/07    3/54
P.NL  000002$ XD    1/04    4/04    4/07
Q00   000000'       1/15    2/29
Q00B  000111'       2/29    2/44
Q01   000017'       1/30    2/30
Q01B  000112'       2/30    3/50
Q02   000025'       1/36    2/31
Q02B  000113'       2/31    3/53
Q03   000051'       1/56    2/32
Q03B  000114'       2/32    2/50
Q05   000070'       2/11    4/11    4/12
Q05B  000223'       3/59    4/11
Q05B4 000224'       4/03    4/12
Q06   000074'       2/15    4/13
Q06B  000225'       4/06    4/13
Q07   000101'       2/20    2/33
Q07B  000115'       2/33    3/06
QRR   000175'       2/58    3/40
QRS   000176'       3/02    3/41
RM000 000116' EN    1/01    2/34    2/35    2/39
RM001 000000' EN    1/13    1/14
RM010 000126'       2/50    3/04
RM020 000142'       3/06    3/14    3/18    3/24
RM030 000161'       3/09    3/26
RM040 000162'       2/60    3/27
RM090 000165'       2/53    3/31
RM0A0 000171'       2/39    3/31    3/36
RM0A1 000172'       2/40    3/32    3/37
RM0A2 000173'       2/41    3/33    3/38
RM0A3 000174'       2/42    3/34    3/39
RM100 000201' EN    2/48    3/30    3/44    3/45    3/49
RM1A3 000222'       3/49    3/57    4/08    4/10
RMCNT 000000- EN    1/10    1/11    3/21    3/60
RMLIM 000177'       3/16    3/42
R.LN  000003$ XD    1/04    2/52    3/08
VB700 000200' XN    1/03    3/43
VBSUB 000200'       2/46    3/43
```

```
 0016 SP000

SP020  000271'        4/28     4/43
SP030  000305'  EN    5/17     5/18     5/20     5/28
SP080  000316'  EN    4/32     5/29     5/39     5/40
SP090  000322'        4/41     5/45
SP091  000327'        5/46     5/51
SP093  000330'        4/22     5/52     6/33
SP0A0  000331'        4/05     5/53    11/12
SP0A1  000332'        4/06     5/54
SP0A2  000333'        4/07     5/55
SP0A3  000334'        4/08     5/56
SP100  000364'  EN    5/47     6/26     6/27     6/29     6/36     7/29     8/41
SP120  000406'        6/44     6/52
SP130  000415'        6/55     7/01
SP140  000416'        6/58     7/02
SP150  000435'        7/09     7/13     7/21
SP160  000440'  EN    6/59     7/26     7/27
SP164  000460'        7/40     7/47
SP170  000470'  EN    7/35     7/50     7/57     7/58     8/36
SP180  000530'        8/14     8/38
SP190  000531'        7/60     8/40
SP400  000537'  EN    8/13     8/47     8/48     8/52
SP410  000556'        8/59     9/11
SP420  000563'        9/13     9/17
SP430  000567'        9/15     9/22
SP450  000600'  EN    5/51     8/40     9/32     9/33     9/35
SP4A0  000574'        8/52     9/22     9/28
SP4A1  000575'        8/53     9/23     9/29
SP4A2  000576'        8/54     9/24     9/30
SP4A3  000577'        8/55     9/17     9/25     9/31     9/35     9/46
SP800  000631'  EN    5/52    10/03    10/04    10/08
SP820  000654'       10/29    10/44
SP830  000670'       10/32    10/42
SP891  000717'       10/25    10/27    10/43    11/01
SP900  000720'       10/56    11/05
SP990  000725'       11/05    11/12
T0     000620'        9/54    10/02
TAB1A  000015$  XD    1/06     5/50     8/03    10/23
TAB1I  000016$  XD    1/06     8/05     8/08     8/30
TABA   000727'  EN    5/49    11/15    11/16    11/17
TABAA  000325'        5/09     5/49
TABQ   000326'        5/10     5/50
TMAXM  000617'  EN    9/11     9/37     9/52     9/53
TMT1   000707'       10/09    10/11    10/13    10/15    10/17    10/19    10/21
                     10/59    10/60
TMT1A  000706'       10/35    10/47    10/59
TNUM   000000-  EN    1/16     1/17     5/14     8/57     9/14     9/39
TNUM0  000040$  XD    1/08
TNUMA  000352'        5/13     6/14
VB1SB  000335'        4/17     5/57
VB200  000335'  XN    1/11     5/57
VB220  000336'  XN    1/11     5/58
VB240  000337'  XN    1/11     5/59
VB2SB  000336'        5/43     5/58
VB3SB  000337'        4/39     5/59
WRN.A  000012$  XD    1/05     4/44
Z0     000000         5/05    11/20    11/22    11/24    11/25    11/29    11/30
                     11/31    11/34    11/36    11/41    11/44    11/46    11/48
                     11/51    11/53    11/54    11/57    11/59    12/03    12/04
```

```
0017 SP000

                12/07   12/09   12/11   12/13   12/15   12/18   12/19
                12/22   12/23   12/25   12/27   12/29   12/34   12/36
                12/38   12/41   12/43   12/49   12/53   12/56   12/57
                12/59   13/02   13/04   13/05
```

```
0001 AL000 MACRO REV 06.50                    19:48:01 10/31/80
                                .TITL   AL000   ;STORAGE
02                      ;
03                              .EXTN   VB300,VB320,VB240
04                              .EXTN   Q01.2,Q07.2,Q08.2
05                      ;
06                              .EXTD   DIG.,GT.CH,R.LN,P.CR,FM.ER,CK.CR
07                              .EXTD   IBUF,MVINI,WRN.A,G.ADR,R.PS
08                              .EXTD   TAB1A,TAB3A,TAB4A,ADR,HIADR
09                              .EXTD   IFMOD,LADR1,LADR2,LINCR,BFPT,ERCT
10                      ;
11          000001              .TXTM   1
12                      ;
13                              .NREL
14                              .ENT    AL001
15                      AL001:
16 000000'005040 Q00:   .TXT    /<12> * * * ALARM LIMITS * * *<15>/
17          025040
18          025040
19          025040
20          040514
21          040522
22          046440
23          046111
24          046511
25          052123
26          020052
27          020052
28          020052
29          006400
30 000016'041517 Q02:   .TXT    /CONTINUING DESTROYS EXISTING LIMITS, /
31          047124
32          044516
33          052511
34          047107
35          020104
36          042523
37          052122
38          047531
39          051440
40          042530
41          044523
42          052111
43          047107
44          020114
45          044515
46          044524
47          051454
48          020000
49 000041'042516 Q06:   .TXT    /ENTER ADRS OR ADRS,MIN,MAX:<15>/
50          052105
51          051040
52          040504
53          051123
54          020117
55          051040
56          040504
57          051123
58          026115
59          044516
60          026115
```

```
        040530
        035015
        000000
000060'046511 Q14:   .TXT    /MIN INVALID<15>/
        047040
        044516
        053101
        046111
        042015
        000000
000067'046501 Q15:   .TXT    /MAX INVALID<15>/
        054040
        044516
        053101
        046111
        042015
        000000
000075'046511 Q16:   .TXT    /MIN > MAX<15>/
        047040
        037040
        046501
        054015
        000000
                     .TITL   AL000    ;PRELIMINARY
                     .ENT    AL000
              ;
              ; * * * ALARM LIMIT SELECTION * * *
              ;
000104'040463 AL000: STA     0,AL0A0
000105'044463        STA     1,AL0A1
000106'050463        STA     2,AL0A2
000107'054463        STA     3,AL0A3
              ;
000110'020475        LDA     0,Q00B
000111'006004$       JSR     @P.CR              ;TYPE "ALARM LIMIT"
              ;
000112'020010$       LDA     0,MVINI
000113'101005        MOV     0,0,SNR
000114'000421        JMP     AL015              ;CHECK MODE ONLY
              ;ASK "INIT,MOD, OR CK?"
000115'005456        JSR     @VB1SB             ;EXPLAIN INIT,MOD,CK
000116'020470 AL010: LDA     0,Q01B
000117'006004$       JSR     @P.CR              ;TYPE "INIT,MOD OR CK?"
              ;
000120'006003$       JSR     @R.LN              ;READ A LINE
000121'002475        JMP     @AL093             ;JUST CR, GO TO NEXT SEC
000122'020007$       LDA     0,IBUF
000123'006002$       JSR     @GT.CH             ;GET CHARACTER
              ;
000124'020452        LDA     0,QRI
000125'122405        SUB     1,0,SNR
000126'000413        JMP     AL020              ;INITIALIZE
              ;
000127'020450        LDA     0,QRM
000130'122405        SUB     1,0,SNR
000131'000427        JMP     AL080              ;MODOFY
              ;
000132'020446        LDA     0,QRC
000133'122404        SUB     1,0,SZR
000134'000762        JMP     AL010              ;TRY AGAIN
```

```
                       ;CHECK ONLY - DON'T HALT PROCESSING
                                                     ;ZERO FLAG FOR NO MODS
000135'040021$AL015:  STA      0.IFMOD               ;EXPLAIN ADRS
000136'006437         JSR      @VB3SB                ;TYPE "ENTER ADR"
000137'020452         LDA      0,Q07B
000140'000424         JMP      AL090
                       ;INITIALIZE
                                                     ;WARNING MESSAGE
000141'020446 AL020:  LDA      0,Q02B                ;CHANCE TO ABORT
000142'006011$        JSR      @WRN.A                ;ABORT
000143'000753         JMP      AL010
                   ;    MIN & MAX TABLES
                      LDA      1,HIADR
000144'024020$        STA      1,LINCR
000145'044024$        LDA      2,TAB3A
000146'030015$        LDA      0,0,2                 ;DEFAULT MIN
000147'021000         LDA      3,TAB4A
000150'034015$        LDA      1,0,3                 ;DEFAULT MAX
000151'025400
                   ;
000152'041000 AL040:  STA      0,0,2
000153'045400         STA      1,0,3
000154'151400         INC      2,2
000155'175400         INC      3,3
000156'014024$        DSZ      LINCR
000157'000773         JMP      AL040
                       ;INIT OR MODIFY
                                                     ;GET CONSTANT 1
000160'102520 AL080:  SUBZL    0,0                   ;SET FLAG TO PERMIT MODS
000161'040021$        STA      0,IFMOD
                   ;                                 ;EXPLAIN AL SELECTION
                      JSR      @VB2SB                ;TYPE "ENTER ADR,MIN,MAX"
000162'006412         LDA      0,Q06B
000163'020425 
000164'006004$AL090:  JSR      @P.CR
000165'105000         MOV      0,1
000166'000431         JMP      AL100
000167'000000 AL0A0:  .-.
000170'000000 AL0A1:  .-.
000171'000000 AL0A2:  .-.
000172'000000 AL0A3:  .-.
000173'177777 VB1SB:  VB300
000174'177777 VB2SB:  VB320
000175'177777 VB3SB:  VB240
                      .TITL    AL002    ;STORAGE
                      .ENT     AL002
                       ;CONSTANTS & STORAGE
                      AL002:
000176'000111 QRI:     "I
000177'000115 QRM:     "M
000200'000103 QRC:     "C
000201'000054 QRCM:    ",
000202'001777 K1777:   1777      ;MAXIMUM RETURNED VALUE
000203'000000 MIN:     .-.
000204'000000 MAX:     .-.
                   ;
000205'000000"Q00B:    Q00*2
000206'177777 Q01B:    Q01.2
000207'000034"Q02B:    Q02*2
000210'000102"Q06B:    Q06*2
000211'177777 Q07B:    Q07.2
000212'177777 Q08B:    Q08.2
000213'000140"Q14B:    Q14*2
000214'000156"Q15B:    Q15*2
000215'000174"Q16B:    Q16*2
```

```
                      ;
000215'000357'AL093: AL900
                     .TITL   AL100   ;PROCESS
                     .ENT    AL100
                  ;ROUTINE TO LOOP FOR ALARM LIMIT MODIFICATION
000217'020773 AL100: LDA     0,Q08B              ;ISSUE PROMPT
000220'006004$       JSR     @P.CR               ;ISSUE PROMPT
                  ;READ LINE - INITIALIZE
000221'006003$       JSR     @R.LN
000222'002774        JMP     @AL093
                  ;GET & CHECK ADR(S)
000223'006012$       JSR     @G.ADR
000224'000773        JMP     AL100               ;ERROR OR REQUEST
000225'126400        SUB     1,1
000226'044530        STA     1,IFDEF             ;INITIALIZE FOR DEFAULT
                  ;GET & CHECK MIN
000227'006001$       JSR     @DIG.               ;CONVERT DIGIT STRING
000230'044753        STA     1,MIN               ;SAVE VALUE
000231'151004        MOV     2,2,SZR             ;CHECK FOR NO DIGITS
000232'000405        JMP     AL120
                  ;    NO MIN: USE PRESENT VALUE
000233'126000        ADC     1,1
000234'044522        STA     1,IFDEF
000235'044746        STA     1,MIN
000236'000411        JMP     AL130
                  ;    CHECK VALUE
000237'030743 AL120: LDA     2,K1777             ;MAX IS 1023
000240'132432        SUBZ#   1,2,SZC
000241'000406        JMP     AL130               ;OK - AC1 <= 1023
                  ;    ERROR
000242'040025$       STA     0,BFPT
000243'020750        LDA     0,Q14B
000244'006004$       JSR     @P.CR               ;TYPE "MIN INVALID"
000245'010026$       ISZ     ERCT
000246'020025$       LDA     0,BFPT
                  ;
                  ;GET & CHECK COMMA
000247'006002$AL130: JSR     @GT.CH
000250'030731        LDA     2,QRCM
000251'132404        SUB     1,2,SZR
000252'006005$       JSR     @FM.ER              ;TYPE "FORMAT ERROR"
                  ;GET & CHECK MAX
000253'006001$       JSR     @DIG.                        -
000254'044730        STA     1,MAX
000255'151004        MOV     2,2,SZR
000256'000405        JMP     AL140
                  ;    NO MAX: USE PRESENT VALUE
000257'126000        ADC     1,1
000260'044476        STA     1,IFDEF
000261'044723        STA     1,MAX
000262'000411        JMP     AL150
                  ;    CHECK VALUE
000263'030717 AL140: LDA     2,K1777             ;MAX IS 1023
000264'132432        SUBZ#   1,2,SZC
000265'000406        JMP     AL150
                  ;    ERROR
000266'040025$       STA     0,BFPT
000267'020725        LDA     0,Q15B              ;TYPE "MAX INVALID"
000270'006004$       JSR     @P.CR
000271'010026$       ISZ     ERCT
```

```
0005 AL000
000272'020025$        LDA     0,BFPT
               ;
               ;GET & CHECK CR
000273'006006$AL150:  JSR     @CK.CR
000274'000401         JMP     .+1
               ;CHECK MIN < MAX
000275'020461         LDA     0,IFDEF          ;CAN'T CHECK IF ONE LEFT INTACT
000276'101004         MOV     0,0,SZR
000277'000410         JMP     AL160
               ;
000300'020703         LDA     0,MIN
000301'024703         LDA     1,MAX
000302'106432         SUBZ#   0,1,SZC
000303'000404         JMP     AL160            ;MIN <= MAX
               ;   ERROR
000304'020711         LDA     0,Q16B
000305'006004$        JSR     @P.CR
000306'010026$        ISZ     ERCT
               ;CHECK FOR ANY ERRORS IN LINE
000307'024026$AL160:  LDA     1,ERCT
000310'125004         MOV     1,1,SZR
000311'000706         JMP     AL100
               ;LOOP TO MODIFY TABLE
000312'020022$        LDA     0,LADR1
               ;   GET TABLE ADDRESSES
000313'034016$AL170:  LDA     3,TAB4A
000314'117000         ADD     0,3
000315'030015$        LDA     2,TAB3A
000316'113000         ADD     0,2
               ;   LOOP
000317'024023$        LDA     1,LADR2
000320'106433         SUBZ#   0,1,SNC
000321'000676         JMP     AL100            ;END OF LOOP
               ;
000322'024434         LDA     1,IFDEF
000323'125004         MOV     1,1,SZR
000324'000406         JMP     AL180
               ;   BOTH SPECIFIED
000325'024656         LDA     1,MIN
000326'045000         STA     1,0,2
000327'024655         LDA     1,MAX
000330'045400         STA     1,0,3
000331'000421         JMP     AL190
               ;   ONE LEFT INTACT
000332'040017$AL180:  STA     0,ADR            ;SAVE FOR R.PS
000333'020650         LDA     0,MIN
000334'101112         MOVL#   0,0,SZC
000335'021000         LDA     0,0,2            ;GET EXISTING MIN
               ;
000336'024646         LDA     1,MAX
000337'125112         MOVL#   1,1,SZC
000340'025400         LDA     1,0,3            ;GET EXISTING MAX
               ;
000341'106432         SUBZ#   0,1,SZC
000342'000405         JMP     AL182            ;OK
               ;   MIN > MAX
000343'020652         LDA     0,Q16B
000344'006004$        JSR     @P.CR
000345'006013$        JSR     @R.PS
000346'000403         JMP     AL185
```

```
0005 AL000
                      ;    SET MIN & MAX
000347'041000 AL182: STA      0,0,2
000350'045400        STA      1,0,3
000351'020022$AL185: LDA      0,LADR1
                      ;    LOOP
000352'024024$AL190: LDA      1,LINCR
000353'123000        ADD      1,0
000354'040022$       STA      0,LADR1
000355'000736        JMP      AL170
000356'000000 IFDEF: .-.
                     .TITL    AL900    ;END
                      ;
                      ;    END OF ALARM LIMIT SELECTION
                      ;
000357'020610 AL900: LDA      0,AL0A0
000360'024610        LDA      1,AL0A1
000361'030610        LDA      2,AL0A2
000362'002610        JMP      @AL0A3
                      ;
                     .TITL    AL000
                     .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
 0007 AL000

ADR    000017$ XD    1/08    5/45
AL000  000104' EN    1/01    2/24    2/25    2/29    5/30
AL001  000000' EN    1/14    1/15
AL002  000175' EN    3/40    3/41    3/43
AL010  000116'       2/42    2/60    3/09
AL015  000135'       2/39    3/02
AL020  000141'       2/52    3/07
AL040  000152'       3/18    3/23
AL080  000160'       2/56    3/25
AL090  000164'       3/05    3/30
AL093  000216'       2/46    4/02    4/10
AL0A0  000167'       2/29    3/33    6/15
AL0A1  000170'       2/30    3/34    6/16
AL0A2  000171'       2/31    3/35    6/17
AL0A3  000172'       2/32    3/36    6/18
AL100  000217' EN    3/32    4/03    4/04    4/06    4/13    5/22    5/33
AL120  000237'       4/20    4/27
AL130  000247'       4/25    4/29    4/38
AL140  000263'       4/46    4/53
AL150  000273'       4/51    4/55    5/04
AL160  000307'       5/09    5/14    5/20
AL170  000313'       5/26    6/09
AL180  000332'       5/37    5/45
AL182  000347'       5/55    6/02
AL185  000351'       5/60    6/04
AL190  000352'       5/43    6/06
AL900  000357'       4/02    6/11    6/15
BFPT   000025$ XD    1/09    4/31    4/35    4/57    5/01
CK.CR  000006$ XD    1/06    5/04
DIG.   000001$ XD    1/06    4/17    4/43
ERCT   000026$ XD    1/09    4/34    4/60    5/18    5/20
FM.ER  000005$ XD    1/06    4/41
GT.CH  000002$ XD    1/06    2/48    4/38
G.ADR  000012$ XD    1/07    4/12
HIADR  000020$ XD    1/08    3/11
IBUF   000007$ XD    1/07    2/47
IFDEF  000356'       4/15    4/23    4/49    5/07    5/35    6/10
IFMOD  000021$ XD    1/09    3/02    3/26
K1777  000202'       3/48    4/27    4/53
LADR1  000022$ XD    1/09    5/24    6/04    6/08
LADR2  000023$ XD    1/09    5/31
LINCR  000024$ XD    1/09    3/12    3/22    6/06
MAX    000204'       3/50    4/44    4/50    5/12    5/41    5/50
MIN    000203'       3/49    4/18    4/24    5/11    5/39    5/46
MVINI  000010$ XD    1/07    2/37
P.CR   000004$ XD    1/06    2/35    2/43    3/30    4/07    4/33    4/59
                     5/17    5/58
Q00    000000'       1/16    3/52
Q00B   000205'       2/34    3/52
Q01B   000206'       2/42    3/53
Q01.2  000206' XN    1/04    3/53
Q02    000016'       1/30    3/54
Q02B   000207'       3/07    3/54
Q06    000041'       1/49    3/55
Q06B   000210'       3/29    3/55
Q07B   000211'       3/04    3/56
Q07.2  000211' XN    1/04    3/56
Q08B   000212'       3/57    4/06
Q08.2  000212' XN    1/04    3/57
```

```
  3000 AL000

Q14    000062'          2/04     3/58
Q14B   000213'          3/58     4/32
Q15    000067'          2/11     3/59
Q15B   000214'          3/59     4/58
Q16    000075'          2/18     3/60
Q16B   000215'          3/60     5/16     5/57
QRC    000200'          2/58     3/46
QRCM   000201'          3/47     4/39
QRI    000176'          2/50     3/44
QRM    000177'          2/54     3/45
R.LN   000003$ XD       1/06     2/45     4/09
R.PS   000013$ XD       1/07     5/59
TAB1A  000014$ XD       1/08
TAB3A  000015$ XD       1/08     3/13     5/28
TAB4A  000016$ XD       1/08     3/15     5/26
VB150  000173'          2/41     3/37
VB240  000175' XN       1/03     3/39
VB255  000174'          3/29     3/38
VB300  000173' XN       1/03     3/37
VB320  000174' XN       1/03     3/38
VB355  000175'          3/03     3/39
WRN.A  000011$ XD       1/07     3/08
```

```
0001 LG000 MACRO REV 06.50              19:59:00 10/31/80
                         .TITL   LG000   ;LOGGING
                         .EXTN   VB400,Q13.2
                         .EXTD   P.NL,P.CR,R.LN.DIG.,BN.AS,GT.CH.MUL.U
                         .EXTD   WRN.A,IBUF
        000001           .TXTM   1

                         .ZREL
                         .ENT    LGMNC,LGMNS,LGMNB
 000000-000002 LGMNC: 2          ;MINUTE COUNTER
 000001-000000 LGMNS: .-.        ;INTERVAL IN MINUTES (LOG CONS)
 000002-000000 LGMNB: .-.        ;INTERVAL IN MINUTES (LOG B CONS)
                         .NREL
                         .ENT    LG001
                   LG001:
 000000'005040 Q00:      .TXT    /<12> * * * LOGGING INTERVAL * * *<15>/
        025040
        025040
        025040
        046117
        043507
        044516
        043440
        044516
        052105
        051126
        040514
        020052
        020052
        020052
        006400
 000020'044516 Q01:      .TXT    /INTERVAL IS /
        052105
        051126
        040514
        020111
        051440
        000000
 000027'044117 Q02:      .TXT    /HOURS.<15>/
        052522
        051456
        006400
 000033'046511 Q03:      .TXT    /MINUTES.<15>/
        047125
        052105
        051456
        006400
 000040'047105 Q04:      .TXT    /NEW INTERVAL: /
        053440
        044516
        052105
        051126
        040514
        035040
        000000
 000050'044117 Q05:      .TXT    /HOURS OR MINUTES: /
        052522
        051440
        047522
        020115
        044516
```

```
       052524
       042523
       035040
       000000
000062'047117 Q06:    .TXT    /NO LOGGING.<15>/
       020114
       047507
       043511
       047107
       027015
       000000
000071'040514 Q07:    .TXT    /ALL VALUES IN VOLTS<15>/
       046040
       053101
       046125
       042523
       020111
       047040
       053117
       046124
       051415
       000000
000104'053101 Q12:    .TXT    /VALUE ERROR<15>/
       046125
       042440
       042522
       051117
       051015
       000000
000113'042101 Q14:    .TXT    /DANGEROUS OPTION: FAILED SENSORS ARE
       047107
       042522
       047525
       051440
       047520
       052111
       047516
       035040
       043101
       044514
       042504
       020123
       042516
       051517
       051123
       020101
       051105
000135'020116  NOT DETECTED.<15>/
       047524
       020104
       042524
       042503
       052105
       042056
       006400
000145'000000 Q15:    .TXT    /<0>/

000146'000000"Q00B:  Q00*2
000147'000040"Q01B:  Q01*2
000150'000056"Q02B:  Q02*2
```

```
000151'000066"Q03B:  Q03*2
000152'000100"Q04B:  Q04*2
000153'000120"Q05B:  Q05*2
000154'000144"Q06B:  Q06*2
                     .TITL   LG000   ;SELECTION
                     .ENT    LG000
                ;
                ;    LOGGING INTERVAL SELECTION
000155'040510 LG000: STA     0,LG0A0          ;SAVE ACCUMULATORS
000156'044510        STA     1,LG0A1
000157'050510        STA     2,LG0A2
000160'054510        STA     3,LG0A3
                ;
000161'020765        LDA     0,Q00B           ;TYPE "LOGGING"
000162'006002$       JSR     @P.CR
000163'006517        JSR     @VBSUB           ;PROMPTING
                ;
000164'004521 LG010: JSR     LG100            ;TYPE PRESENT INTERVAL
                ;
000165'020765 LG015: LDA     0,Q04B           ;TYPE "ENTER"
000166'006002$       JSR     @P.CR
000167'006003$       JSR     @R.LN            ;READ RESPONSE
000170'000471        JMP     LG090            ;JUST CR:  RETURN
                ;
000171'020011$       LDA     0,IBUF           ;DECODE NUMBER
000172'006004$       JSR     @DIG.
000173'040504        STA     0,LG091          ;SAVE BYTE POINTER
                ;CHECK FOR 1 OR 2 DIGITS
000174'034504        LDA     3,K2             ;CONSTANT 2
000175'151004        MOV     2,2,SZR
000176'156433        SUBZ#   2,3,SNC
000177'000543        JMP     LG210            ;FORMAT ERROR
                ;SAVE VALUE
000200'044501        STA     1,LG093
000201'125005        MOV     1,1,SNR
000202'000446        JMP     LG070            ;CANCEL LOGGING
                ;CHECK NEXT CHAR FOR BLANK
000203'020474        LDA     0,LG091
000204'006006$       JSR     @GT.CH           ;GET NEXT CHAR
000205'030467        LDA     2,QRBL
000206'132404        SUB     1,2,SZR
000207'000402        JMP     LG030            ;NOT BLANK, CHECK H OR M
                ;CHECK NEXT CHAR FOR H OR M
000210'006006$LG020: JSR     @GT.CH           ;GET NEXT CHAR
                ;    CHECK FOR H
000211'030464 LG030: LDA     2,QRH
000212'132405        SUB     1,2,SNR
000213'000412        JMP     LG040            ;PROCESS HOURS
                ;    CHECK FOR M
000214'030462        LDA     2,QRM
000215'132405        SUB     1,2,SNR
000216'000415        JMP     LG050            ;PROCESS MINUTES
                ;    ERROR - NOT HOURS OR MINUTES
000217'020734        LDA     0,Q05B           ;TYPE "HOURS OR MINS?"
000220'006002$       JSR     @P.CR
000221'006003$       JSR     @R.LN            ;READ RESPONSE
000222'000742        JMP     LG010            ;JUST CR: TRY AGAIN
000223'020011$       LDA     0,IBUF
000224'000764        JMP     LG020
                ;HOURS
```

```
                                .ENT    LG040
   000225'024454 LG040: LDA     1,LG093
   000226'044444        STA     1,LGHR
   000227'030444        LDA     2,D60
   000230'006007$       JSR     @MUL.U          ;TIMES 60 FOR
   000231'044001-       STA     1,LGMNS         ;INTERVAL IN MINUTES
   000232'000413        JMP     LG060
                 ;MINUTES
   000233'024446 LG050: LDA     1,LG093
                 ;
   000234'030437        LDA     2,D60
   000235'146432        SUBZ#   2,1,SZC
   000236'000507        JMP     LG220           ;VALUE ERROR
   000237'030432        LDA     2,LGMIN
   000240'146433        SUBZ#   2,1,SNC
   000241'000504        JMP     LG220           ;VALUE ERROR
                 ;
   000242'044001-       STA     1,LGMNS         ;INTERVAL IN MINUTES
   000243'126400        SUB     1,1
   000244'044426        STA     1,LGHR
                 ;COMMON PROCESSING - INITIALIZE LOGGING
   000245'024433 LG060: LDA     1,K2
   000246'044000-       STA     1,LGMNC         ;MINUTE COUNT = 2
   000247'000411        JMP     LG080
                 ;NO LOGGING
   000250'044422 LG070: STA     1,LGHR
   000251'044001-       STA     1,LGMNS
   000252'044000-       STA     1,LGMNC
                 ;   WARNING
   000253'020430        LDA     0,Q14B          ;PRINT WARNING
   000254'006002$       JSR     @P.CR
   000255'020427        LDA     0,Q15B          ;CHANCE TO ABORT
   000256'006010$       JSR     @WRN.A
   000257'000706        JMP     LG015
                 ;RETURN FROM THIS ROUTINE
   000260'004425 LG080: JSR     LG100           ;TYPE PRESENT INTERVAL
   000261'020404 LG090: LDA     0,LG0A0
   000262'024404        LDA     1,LG0A1
   000263'030404        LDA     2,LG0A2
   000264'002404        JMP     @LG0A3
                 ;
   000265'000000 LG0A0: .-.
   000266'000000 LG0A1: .-.
   000267'000000 LG0A2: .-.
   000270'000000 LG0A3: .-.
                        .ENT     LGMIN
   000271'000002 LGMIN: 2.       ;MINIMUM INTERVAL
   000272'000001 LGHR:  1
   000273'000074 D60:   60.
   000274'000040 QRBL:  40
   000275'000110 QRH:   "H
   000276'000115 QRM:   "M
   000277'000000 LG091: .-.      ;BUFFER BYTE POINTER
   000300'000002 K2:    2
   000301'000000 LG093: .-.      ;TEMP STORAGE
   000302'177777 VSSUB: VS400
   000303'000226"Q14B:  Q14*2
   000304'000312"Q15B:  Q15*2
                        .TITL    LG100   ;PRESENT STATUS
                        .ENT     LG100
                        ;
                        ;ROUTINE TO TYPE PRESENT INTERVAL
                        ;
000305'054426 LG100:  STA     3,LG1A3
000306'024754         LDA     1,LGHR          ;CHECK HOURS
000307'125004         MOV     1,1,SZR
000310'000407         JMP     LG110
000311'024001-        LDA     1,LGMNS         ;CHECK MINUTES
000312'125004         MOV     1,1,SZR
000313'000404         JMP     LG110
                        ;
000314'020640         LDA     0,Q06B          ;PRINT "NO LG"
000315'006002$        JSR     @P.CR
000316'000414         JMP     LG190
                        ;CONVERT TO ASCII
000317'020421 LG110:  LDA     0,Q08B          ;BYTE POINTER
000320'006005$        JSR     @BN.AS          ;CONVERT
                        ;PRINT
000321'020626         LDA     0,Q01B          ;TYPE "PRESENT INTERVAL"
000322'006002$        JSR     @P.CR
000323'020416         LDA     0,Q08B4         ;TYPE 2 DIGITS
000324'006001$        JSR     @P.NL
000325'020623         LDA     0,Q02B          ;HOURS
000326'024744         LDA     1,LGHR
000327'125005         MOV     1,1,SNR
000330'020621         LDA     0,Q03B          ;MINUTES
000331'006001$        JSR     @P.NL           ;TYPE UNITS
                        ;RETURN
000332'002401 LG190:  JMP     @LG1A3
                        ;STORAGE
000333'000000 LG1A3:  .-.                     ;RETURN ADDRESS
000334'054130 Q08:    .TXT    /XXXXXX /
       054130
       054130
       020000
000340'000670"Q08B:   Q08*2
000341'000674"Q08B4:  Q08*2+4
                        ;ERROR ROUTINES
000342'020406 LG210:  LDA     0,Q13B          ;TYPE "FORMAT ERROR"
000343'006002$        JSR     @P.CR
000344'000621         JMP     LG015
                        ;
000345'020404 LG220:  LDA     0,Q12B          ;TYPE "VALUE ERROR"
000346'006002$        JSR     @P.CR
000347'000615         JMP     LG015
000350'177777 Q13B:   Q13.2
000351'000210"Q12B:   Q12*2
                      .TITL   LG500   ;LOGGING TASK
                      .ENT    LG500
                      .EXTN   .KILL,.REC
                      .EXTD   HIADR,TAB1A,TLGA,LGSAV,P.VAL,P.DT
                      .EXTD   LINCR,ADR,IFLOG,IFULT
                        ;   LOGGING TASK
                        ;
000352'102520 LG500:  SUBZL   0,0
000353'040022$        STA     0,IFLOG         .CHECK IF LOGGING REQUESTED
000354'020001-        LDA     0,LGMNS
000355'101004         MOV     0,0,SZR
000356'000404         JMP     LG510
```

```
000357'020002-             LDA     0,LGMNB           ;CHECK LOGGING B
000360'101005              MOV     0,0,SNR
000361'177777              .KILL                     ;IF NOT, KILL
                   ;WAIT FOR DATA TO BE READY
000362'020014$LG510:       LDA     0,TLGA
000363'040015$             STA     0,LGSAV
000364'034460              LDA     3,LGMSA
000365'102400              SUB     0,0               ;WIPE OUT ANY
000366'041400              STA     0,0,3             ;MESSAGE WAITING
000367'161000              MOV     3,0               ;WAIT FOR NEXT
000370'177777              .REC
000371'102400              SUB     0,0
000372'040015$             STA     0,LGSAV           ;SUPPRESS LOGGING SAVE
                   ;PRINT HEADER
000373'020453              LDA     0,QLCR
000374'006017$             JSR     @P.DT             ;PRINT DATE & TIME
000375'020023$             LDA     0,IFVLT
000376'101005              MOV     0,0,SNR
000377'000403              JMP     LG515
000400'020447              LDA     0,Q07B
000401'006002$             JSR     @P.CR             ;PRINT "ALL IN VOLTS"
                   ;LOOP TO PRINT ALL VALUES
000402'020437 LG515:       LDA     0,LOADR           ;LOW ADR VARIABLE
000403'040021$             STA     0,ADR
000404'024012$             LDA     1,HIADR
000405'106400              SUB     0,1
000406'044020$             STA     1,LINCR
                   ;LOOP TO PRINT LINE
000407'024021$LG520:       LDA     1,ADR
                   ;CHECK RATE
000410'034013$             LDA     3,TAB1A
000411'137000              ADD     1,3
000412'021400              LDA     0,0,3
000413'030432              LDA     2,MKRAT
000414'113405              AND     0,2,SNR
000415'000420              JMP     LG540             ;LOOP INCR
                   ;CHECK LOGGING "B"
000416'030002-             LDA     2,LGMNB
000417'151005              MOV     2,2,SNR
000420'000404              JMP     LG530             ;LOGGING A, GO PRINT
                   ;
000421'030421              LDA     2,MKLGB           ;LOGGING B, CHECK IF SELECTED
000422'113405              AND     0,2,SNR
000423'000412              JMP     LG540
                   ;CONVERT & PRINT ADR
000424'020424 LG530:       LDA     0,Q50B
000425'006005$             JSR     @BN.AS
000426'020423              LDA     0,Q50B3
000427'006002$             JSR     @P.CR
                   ;GET VALUE
000430'030021$             LDA     2,ADR
000431'034014$             LDA     3,TLGA
000432'157000              ADD     2,3
000433'025400              LDA     1,0,3             ;GET VALUE
000434'006016$             JSR     @P.VAL
                   ;INCR LOOP
000435'010021$LG540:       ISZ     ADR
000436'014020$             DSZ     LINCR
000437'000750              JMP     LG520
                   ;****PRINT CALCS
```

```
0007 LG000
000440'000722         JMP     LG510
                 ;
                      .ENT    LOADR,LGMSG
000441'000001 LOADR:  1       ;CHANGE FOR TESTING
000442'020000 MKLGB:  1B2
000443'000000 LGMSG:  .-.
000444'000443'LGMSA:  LGMSG
000445'016000 MKRAT:  7B5
000446'006400 QLCR:   6400
000447'000162"Q07B:   Q07*2
000450'001124"Q50B:   Q50*2
000451'001127"Q50B3:  Q50*2+3
                 ;
000452'054130 Q50:    .TXT    /XXXADR /
       054101
       042122
       020000
                      .TITL   LG000
                      .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
0001 TR000 MACRO REV 06.50                    20:21:12 10/31/80
                             .TITL   TR000   .PAGE 1
                             .ENT    TR001,MT0A,MT1A,MTNDV
                             .ENT    MTC,MTN,MTCS,MTNS,MTCCN,MTNCN
                             .ENT    MTFLG,MTTK1
                             .EXTN   .REC,.TIDK,.TASK,.KILL,VB800,VB920
                             .EXTD   P.CR,P.NL,R.LN,MUINI
                             .EXTD   MT.AS,MT.DV,MT.UN,MT.HD,MT.EF,MT.RL
                             .ZREL
   000000-000000 MTC:        .-.     ;NEXT WORD, CURR BUFF
   000001-000000 MTN:        .-.     ;NEXT WORD, NON-CURR BUFF
   000002-000000 MTCS:       .-.     ;START CURR BUFF DATA
   000003-000000 MTNS:       .-.     ;START NON-CURR BUFF DATA
   000004-000000 MTCCN:      .-.     ;CURR CHANNEL
   000005-000000 MTNCN:      .-.     ;NON-CURR CHANNEL
   000006-000000 MTFLG:      .-.     ;-1 RECORDING, 0 STANDBY, N WRITE N RECS
   000007-000000 MTTK1:      .-.     ;MESSAGE LOC
   000010-000000 MTNDV:      .-.     ;NUMBER OF MT DEVICES (1 OR 2)
   000011-000012'MT0A:       MT0T    ;ADDRESS OF MT0 TABLE
   000012-000015'MT1A:       MT1T    ;ADDRESS OF MT1 TABLE
          000001             .TXTM   1
                             .NREL
                      TR001:
                      ;STORAGE - OCCURS ONCE
   000000'046524 MT0:        .TXT    /MT0/
          030000
   000002'046524 MT1:        .TXT    /MT1/
          030400
   000004'046524 MT0.0:      .TXT    /MT0:0/
          030072
          030000
   000007'046524 MT1.0:      .TXT    /MT1:0/
          030472
          030000
                      ;
   000012'000000 MT0T:       0
   000013'000000"            MT0*2
   000014'000010"            MT0.0*2
   000015'000001 MT1T:       1
   000016'000004"            MT1*2
   000017'000016"            MT1.0*2
                      ;
   000020'005040 000:        .TXT    /<12> * * * TAPE UNIT * * *<15>/
          025040
          025040
          025040
          052101
          050105
          020125
          047111
          052040
          025040
          025040
          025015
          000000
   000035'030440 001:        .TXT    /1 OR 2 TAPE UNITS: /
          047522
          020062
          020124
          040520
          042440
```

```
0002 TR000
       052516
       044524
       051472
       020000
000047'046517 Q02:    .TXT    /MOUNT TAPE FOR WRITING./
       052516
       052040
       052101
       050105
       020106
       047522
       020127
       051111
       052111
       047107
       026000
000053'046517 Q03:    .TXT    /MOUNT TAPE FOR ALTERNATE./
       052516
       052040
       052101
       050105
       020106
       047522
       020101
       046124
       042522
       047101
       052105
       026000
000100'052131 Q07:    .TXT    /TYPE UNIT TO RELEASE (0 OR 1): /
       050105
       020125
       047111
       052040
       052117
       020122
       042514
       042501
       051505
       020050
       030040
       047522
       020061
       024472
       020000
000120'005105 Q08:    .TXT    /<12>EOT<15>/
       047524
       005400
000123'030440 Q51:    .TXT    /1 UNIT/
       052516
       044524
       000000
000127'031040 Q52:    .TXT    /2 UNITS/
       052516
       044524
       051400
000133'026040 Q54:    .TXT    /, 0 NOT READY/
       030040
       047117
       052040
       051105
       040504
       054400
000142'026040 Q55:   .TXT    /, ALT READY/
       040514
       052040
       051105
       040504
       054400
000150'026040 Q56:   .TXT    /, ALT NOT READY/
       040514
       052040
       047117
       052040
       051105
       040504
       054400
000160'027015 Q57:   .TXT    /.<15>/
       000000
                     .TITL   TR000   ;PAGE 2
                     .ENT    TR000
              ;*** TAPE UNIT SELECTION - INITIAL
              ;
000162'040422 TR000: STA     0,TR0A0         ;SAVE REGISTERS
000163'044422        STA     1,TR0A1
000164'050422        STA     2,TR0A2
000165'054422        STA     3,TR0A3
              ;TYPE IDENT & EXPLAIN, IF NECESSARY
000166'020422        LDA     0,Q00B          ;TYPE "TAPE UNIT"
000167'006001$       JSR     @P.CR
000170'006422        JSR     @VB0SB          ;TYPE VERBOSE
              ;INITIALIZE CHANNELS
000171'102400        SUB     0,0
000172'040004-       STA     0,MTCCN
000173'040005-       STA     0,MTNCN
              ;GET # DEVICES
000174'020415 TR010: LDA     0,Q01B          ;TYPE "HOW MANY:"
000175'006001$       JSR     @P.CR
000176'006003$       JSR     @R.LN           ;READ RESPONSE
000177'000775        JMP     TR010           ;JUST CR - TRY AGAIN
              ;
000200'006006$       JSR     @MT.DV          ;DECODE #
000201'000773        JMP     TR010           ;ERROR - TRY AGAIN
000202'044010-       STA     1,MTNDV         ;SAVE #
000203'000467        JMP     TR160
              ;
000204'000000 TR0A0: .-.
000205'000000 TR0A1: .-.
000206'000000 TR0A2: .-.
000207'000000 TR0A3: .-.
000210'000040"Q00B:  Q00*2
000211'000072"Q01B:  Q01*2
000212'177777 VB0SB: VB800
000213'177777 VB2SB: VB820
                     .TITL   TR100   ;PAGE 3
                     .ENT    TR100,TR130
              ;* * * TAPE UNIT SELECTION - OPERATOR REQUESTED
              ;
000214'040770 TR100: STA     0,TR0A0
000215'044770        STA     1,TR0A1
```

```
0004 TR000
   000216'050770          STA     2,TR0A2
   000217'054770          STA     3,TR0A3
                  ;TYPE IDENT & EXPLAIN, IF NECESSARY
   000220'020770          LDA     0,000B            ;TYPE "TAPE UNIT"
   000221'006001$         JSR     @P.CR
   000222'006771          JSR     @VB2SB            ;VERBOSE PROMPTING
                  ;TYPE PRESENT STATUS
   000223'004471          JSR     TR500
                  ;GET # DEVICES
   000224'020765 TR110:   LDA     0,001B            ;TYPE "HOW MANY:"
   000225'006001$         JSR     @P.CR
   000226'006003$         JSR     @R.LN             ;READ RESPONSE
   000227'000433          JMP     TR150             ;JUST CR - LEAVE ALONE
                 ;
   000230'006006$         JSR     @MT.DV            ;DECODE #
   000231'000773          JMP     TR110             ;ERROR - TRY AGAIN
                 ;
   000232'125235          MOVZR#  1,1,SNR           ;CHECK HOW MANY
   000233'000403          JMP     TR120             ;ONE UNIT
                  ;TWO UNITS
   000234'044010-         STA     1,MTNDV
   000235'000425          JMP     TR150
                  ;ONE UNIT
   000236'020010-TR120:   LDA     0,MTNDV           ;CHECK PRESENT VALUE
   000237'101222          MOVZR   0,0,SZC
   000240'000422          JMP     TR150             ;ALREADY 1
                 ;      CHECK NON-CURRENT
   000241'030005-         LDA     2,MTNCN           ;CHECK NON-CURRENT
   000242'151005          MOV     2,2,SNR
   000243'000415          JMP     TR140             ;INACTIVE
                  ;ACTIVE
   000244'020004$         LDA     0,MVINI
   000245'101004          MOV     0,0,SZR
   000246'000403          JMP     TR130
                 ;      RECORDING, RELEASE NON-CURRENT
                 ;         ***CHECK HARDWARE STATUS, NON-CURRENT
   000247'006012$         JSR     @MT.RL
   000250'000410          JMP     TR140
                 ;      NO RECORDING - ASK WHICH TO RELEASE
                 TR130:
                 ;         ***CHECK HARDWARE STATUS, BOTH
   000251'020442 TR132:   LDA     0,007B            ;TYPE "WHICH TO RELEASE"
   000252'006001$         JSR     @P.CR             ;READ RESPONSE
   000253'006003$         JSR     @R.LN             ;JUST CR - ASK AGAIN HOW MANY
   000254'000750          JMP     TR110
                 ;
   000255'006007$         JSR     @MT.UN            ;DECODE UNIT
   000256'000773          JMP     TR132             ;ILLEGAL UNIT - ASK AGAIN WHICH
                 ;
   000257'006012$         JSR     @MT.RL            ;CLOSE & RELEASE ACTIVE DRIVE
                 ;      SET MTNDV=1
   000260'102520 TR140:   SUBZL   0,0
   000261'040010-         STA     0,MTNDV
                 ;
                  ;CHECK STATUS
   000262'020005-TR150:   LDA     0,MTNCN
   000263'101004          MOV     0,0,SZR
   000264'000421          JMP     TR190
                 ;
   000265'020004-         LDA     0,MTCCN
```

```
000266'101005        MOV     0,0,SNR
000267'000403        JMP     TR160
                    ;TYPE WHICH UNIT IN USE
000270'004424        JSR     TR500
000271'000405        JMP     TR170
                    ;BOTH INACTIVE
000272'020417 TR160: LDA     0,Q02B
000273'006005$       JSR     @MT.AS
000274'000776        JMP     .-2
000275'050004-       STA     2,MTCCN
                    ;CHECK 1 OR 2 UNITS
000276'020010-TR170: LDA     0,MTNDV
000277'101225        MOVZR   0,0,SNR
000300'000405        JMP     TR190
                    ;TWO UNITS, INITIALIZE SECOND
000301'020411        LDA     0,Q03BB
000302'006005$       JSR     @MT.AS
000303'152400        SUB     2,2                        ;JUST CR
000304'050005-       STA     2,MTNCN
                    ;RETURN
000305'020677 TR190: LDA     0,TR0A0
000306'024677        LDA     1,TR0A1
000307'030677        LDA     2,TR0A2
000310'002677        JMP     @TR0A3
                    ;
000311'000116"Q02B:  Q02*2
000312'000146"Q03BB: Q03*2
000313'000200"Q07B:  Q07*2
                     .TITL   TR500
                     .ENT    TR500
                    ;PRINT STATUS OF TAPE DRIVES
000314'040440 TR500: STA     0,TR5A0
000315'044440        STA     1,TR5A1
000316'050440        STA     2,TR5A2
000317'054440        STA     3,TR5A3
                    ;NUMBER OF UNITS
000320'020441        LDA     0,Q51B
000321'030010-       LDA     2,MTNDV
000322'151233        MOVZR#  2,2,SNC
000323'020437        LDA     0,Q52B
000324'006001$       JSR     @P.CR
                    ;FIRST UNIT
000325'020437        LDA     0,Q54B
000326'034004-       LDA     3,MTCCN
000327'175005        MOV     3,3,SNR
000330'000406        JMP     TR520
                    ;    READY - ASSUME WRITING
000331'020432        LDA     0,Q53B1
000332'024426        LDA     1,QB0
000333'035400        LDA     3,0,3
000334'167000        ADD     3,1
000335'044434        STA     1,Q53+1
                    ;    CHECK IF TWO
000336'151235 TR520: MOVZR#  2,2,SNR
000337'000406        JMP     TR590
000340'006002$       JSR     @P.NL
                    ;SECOND UNIT
000341'020424        LDA     0,Q55B
000342'034005-       LDA     3,MTNCN
000343'175005        MOV     3,3,SNR
```

```
000344'020422         LDA     0,Q56B
000345'006002$TR500:  JSR     @P.NL
                ;TYPE CR
000346'020421         LDA     0,Q57B
000347'006002$        JSR     @P.NL
                ;RETURN
000350'020404         LDA     0,TR5A0
000351'024404         LDA     1,TR5A1
000352'030404         LDA     2,TR5A2
000353'002404         JMP     @TR5A3
                ;STORAGE
000354'000000 TR5A0:  .-.
000355'000000 TR5A1:  .-.
000356'000000 TR5A2:  .-.
000357'000000 TR5A3:  .-.
000360'020060 Q20:    20060
000361'000246"Q51B:   Q51*2
000362'000256"Q52B:   Q52*2
000363'000761"Q53B1:  Q53*2+1
000364'000266"Q54B:   Q54*2
000365'000304"Q55B:   Q55*2
000366'000320"Q56B:   Q56*2
000367'000340"Q57B:   Q57*2
000370'020054 Q53:    .TXT    / , 0 WRITING/
       020060
       020127
       051111
       052111
       047107
       000000
                      .TITL   TR200
                      .ENT    TR200,TPOVF
                      .EXTN   MTBHL
                      .EXTD   S.PUS,P.ER
                ;
                ;   TAPE WRITE
                ;
000377'020530 TR200:  LDA     0,AMTK1
000400'177777         .REC                        ;WAIT FOR MESSAGE
000401'034003-        LDA     3,MTNS
000402'045776         STA     1,-2,3              ;SAVE HALF SEC FLAG
                ;GET TIME OF DAY
000403'006017         .SYSTEM
000404'021003         .GTOD
000405'000000         .-.                         ;ERROR RETURN NOT USED
000406'034003-        LDA     3,MTNS              ;GET START BUFF
000407'041775         STA     0,-3,3              ;SECONDS
000410'045774         STA     1,-4,3              ;MINUTES
000411'051773         STA     2,-5,3              ;HOURS
                ;CHECK IF WRITING SELECTED
000412'030006-        LDA     2,MTFLG
000413'151005         MOV     2,2,SNR
000414'000406         JMP     TR205               ;ZERO - STANDBY
                ;   RECORDING ALREADY
000415'051777         STA     2,-1,3
000416'151123         MOVZL   2,2,SNC
000417'014006-        DSZ     MTFLG
000420'000401         JMP     .+1
000421'000407         JMP     TR210
                ;   MAY BE TIME TO LOG
```

```
 0007 TR000
000422'101005 TR205: MOV     0,0,SNR          ;CHECK SEC 0
000423'125004        MOV     1,1,SZR          ;CHECK MIN 0
000424'000753        JMP     TR200
              ;   YUP - BOTH ZERO
000425'131400        INC     1,2              ;GENERATE 1
000426'050006-       STA     2,MTFLG
000427'051777        STA     2,-1,3
              ;GET DATE
000430'006017 TR210: .SYSTEM
000431'021006        .GDAY
000432'000000        .-.                      ;ERROR RETURN NOT USED
000433'034003-       LDA     3,MTNS           ;GET START BUFF
000434'041772        STA     0,-6,3           ;DAY
000435'045771        STA     1,-7,3           ;MONTH
000436'051770        STA     2,-8.,3          ;YEAR
              ;DON'T WRITE EMPTY BUFFER
000437'024001-       LDA     1,MTN
000440'166405        SUB     3,1,SNR          ;BUFFER LENGTH
000441'000736        JMP     TR200
              ;CHECK BUFFER LENGTH
000442'020470        LDA     0,BUFL
000443'106433        SUBZ#   0,1,SNC
000444'000405        JMP     TR220
              ;   BUFFER TOO LONG
000445'010456        ISZ     TPOVF            ;COUNT ERRORS
000446'000401        JMP     .+1
000447'024462        LDA     1,MTHL
000450'124400        NEG     1,1              ;GET -8 TO MAKE ZERO
              ;WRITE TO TAPE
                     .ENT    TR220
000451'030004-TR220: LDA     2,MTCCN          ;GET CURR CHANNEL #
000452'151005        MOV     2,2,SNR
000453'000724        JMP     TR200            ;CAN'T WRITE
              ;   COMPUTE COMMAND WORD
000454'020454        LDA     0,MTWRI          ;GET COMMAND
000455'107000        ADD     0,1              ;ADD TO DATA LENGTH
000456'034453        LDA     3,MTHL           ;GET HEADER LENGTH
000457'157000        ADD     3,1              ;ADD HEADER LENGTH
              ;   GET WRITE ADDRESS
000460'020003-       LDA     0,MTNS           ;START OF DATA
000461'162400        SUB     3,0              ;SUB HEADER LENGTH
              ;   WRITE TO TAPE
000462'031000        LDA     2,0,2
000463'006017        .SYSTEM
000464'023077        .MTDIO  77
000465'000402        JMP     TR240            ;ERROR RETURN
000466'000711        JMP     TR200
                     .TITL   TR240
                     .ENT    TR240
                     .EXTD   MT.ER
              ;
              ;   ERROR RETURN FROM WRITE - PROCESS EOT
              ;
000467'050450 TR240: STA     2,MTSW           ;SAVE STARUS WORD
000470'024451        LDA     1,B6             ;CHECK EOT BIT
000471'147405        AND     2,1,SNR
000472'006015$       JSR     @MT.ER           ;
              ;SWAP CHANNELS
000473'030005-       LDA     2,MTNCN
000474'151004        MOV     2,2,SZR
```

```
0009 TR000
000475'000416          JMP     TR250           ;ALT CHAN READY
               ;NO ALT CHANNEL AVAILABLE - FINISH CURRENT TAPE
000476'030004-         LDA     2,MTCCN
000477'004545          JSR     MTEND           ;EOF,RELE,P,ER,ALARM "C"
000500'004442          JSR     TR300
                 ;     RESET BUFFER POINTERS
                 ;     DO TAPE REQUEST, INIT AS SUBROUTINE
000501'020437          LDA     0,Q02BB
000502'006005$         JSR     @MT.AS
000503'000775          JMP     .-2
000504'050004-         STA     2,MTCCN         ;ACTIVATE
000505'006010$         JSR     @MT.HD          ;WRITE HEADER
                 ;     IF 2 DEV, TASK TO GET ANOTHER
000506'020010-         LDA     0,MTNDV
000507'101224          MOVZR   0,0,SZR
000510'000412          JMP     TR260
                 ;     ELSE, RETURN
000511'004472          JSR     TR350           ;RESTART KILLED TASKS
000512'000414          JMP     TR290
                       .ENT    TR250
               ;START I/O ON ALT CHANNEL
000513'024004-TR250:   LDA     1,MTCCN         ;FLIP CHAANELS
000514'044005-         STA     1,MTNCN
000515'050004-         STA     2,MTCCN
                 ;     WRITE HEADER
000516'006010$         JSR     @MT.HD
000517'004423          JSR     TR300           ;KILL COMPETING TASKS
                 ;     TASK TO FINISH OLD & GET ANOTHER
000520'024522          LDA     1,TR4A1
000521'000402          JMP     TR270
               ;TASK REQUEST ANOTHER
000522'024517 TR260:   LDA     1,TR4A2
000523'020515 TR270:   LDA     0,TR4WD
000524'177777          .TASK
000525'000401          JMP     .+1
               ;START RECORDING
000526'000651 TR290:   JMP     TR200
                 ;
000527'000007-AMTK1:   MTTK1   ;TASK MESSAGE ADDRESS
000530'050000 MTWRI:   5B3
000531'177777 MTHL:    MTBHL   ;BUFF HEADER LENGTH
000532'007770 BUFL:    4096.-8.
000533'000000 TPOVF:   .-.     ;COUNT ERRORS
000534'000535'AALMC:   .+1
000535'020000          1B2
000536'000000          0
000537'000000 MTSW:    .-.     ;SAVE STATUS WORD
000540'000116"Q02BB:   Q02*2
000541'001000 B6:      1B6     ;EOT BIT
                       .TITL   TR300
                       .ENT    TR300,TR350
                       .EXTN   MN500,.TIDR,.ABORT
               ;SUBROUTINE TO KILL TYPING TASKS & REMEMBER WHICH
000542'040430 TR300:   STA     0,TR3A0
000543'044430          STA     1,TR3A1
000544'050430          STA     2,TR3A2
000545'054430          STA     3,TR3A3
               ;KILL OPERATOR INTERVENTION TASK 3
000546'024431          LDA     1,TK3ID
000547'177777          .ABORT
```

```
0003 TR000
000550'000403        JMP     TR310
000551'044425        STA     1,KIL3
000552'000412        JMP     TR330
                    ;NOT OPERATOR REQUEST
000553'126400 TR310: SUB     1,1
000554'044422        STA     1,KIL3
                    ;KILL TAPE REQUEST TASK 4
000555'024423        LDA     1,TK4ID
000556'000547'       .ABORT
000557'000402        JMP     TR320
000560'000404 .      JMP     TR330
                    ;KILL LOGGING/MONITORING TASK 6
000561'024420 TR320: LDA     1,TK6ID
000562'177777        .TIDK
000563'000403        JMP     TR340
                    ;KILLED SOMETHING - TYPE "EOT"
000564'020416 TR330: LDA     0,Q08B
000565'006001$       JSR     @P.CR
                    ;RETURN
000566'020404 TR340: LDA     0,TR3A0
000567'024404        LDA     1,TR3A1
000570'030404        LDA     2,TR3A2
000571'002404        JMP     @TR3A3
                    ;
000572'000000 TR3A0: .-.
000573'000000 TR3A1: .-.
000574'000000 TR3A2: .-.
000575'000000 TR3A3: .-.
000576'000000 KIL3:  .-.
000577'000003 TK3ID: 3
000600'000004 TK4ID: 4
000601'000006 TK6ID: 6
000602'000240"Q08B:  Q08*2
                    ;
                    ;SUBROUTINE TO REVIVE WHATEVER NECESSARY
000603'054772 TR350: STA     3,TR3A3
000604'050770        STA     2,TR3A2
000605'044766        STA     1,TR3A1
000606'040764        STA     0,TR3A0
                    ;CHECK WHAT TO REVIVE
000607'020767        LDA     0,KIL3
000610'101005        MOV     0,0,SNR
000611'000405        JMP     TR360
                    ;TASK 3 KILLED - READY TASK 0
000612'126400        SUB     1,1
000613'177777        .TIDR                         ;SUSPENDED WHEN 3 CREATED
000614'000400        JMP     .                     ;****HANG FOR DEBUG
000615'000405        JMP     TR390
                    ;TASK 0 OK - START MONITORING/LOGGING
000616'020410 TR360: LDA     0,TK6WD
000617'024410        LDA     1,TK6AD
000620'000524'       .TASK
000621'000400        JMP     .                     ;****HANG FOR DEBUG
                    ;
000622'020750 TR390: LDA     0,TR3A0
000623'024750        LDA     1,TR3A1
000624'030750        LDA     2,TR3A2
000625'002750        JMP     @TR3A3
                    ;
000626'003006 TK6WD: 6B7+6
```

```
 0010 TR000
 000627'177777 TK6AD: MN500
                      .TITL   TR400
                      .ENT    TR400
               ;
               ;   TASK TO FINISH OLD TAPE
               ;
 000630'030005-TR400: LDA     2,MTNCN
 000631'004413        JSR     MTEND            ;EOF,RELE,P.ER,ALARM "C"
               ;
               ;   TASK TO REQUEST A SECOND TAPE
               ;
 000632'020411 TR410: LDA     0,Q03B
 000633'006005$       JSR     @MT.AS
 000634'152400        SUB     2,2              ;CR ONLY
 000635'050005-       STA     2,MTNCN
               ;   RESTART CANCELLED TASK
 000636'004745        JSR     TR350
 000637'177777        .KILL
               ;
 000640'002004 TR4WD: 4B7+4
 000641'000632'TR4A2: TR410
 000642'000630'TR4A1: TR400
 000643'000146"Q03B:  Q03*2
                      .TITL   MTEND
 000644'054411 MTEND: STA     3,MTEA3
 000645'006011$       JSR     @MT.EF
 000646'020666        LDA     0,AALMC
 000647'060277        INTDS
 000650'006013$       JSR     @S.PUS
 000651'060177        INTEN
 000652'006012$       JSR     @MT.RL
 000653'006014$       JSR     @P.ER
 000654'002401        JMP     @MTEA3
 000655'000000 MTEA3: .-.
                      .TITL   TR000
                      .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
 0011 TR000

AALMC 000534'         8/44    10/27
AMTK1 000527'         6/39     8/39
BE    000541'         7/55     8/49
BUFL  000532'         7/21     8/42
KIL3  000576'         9/02     9/06     9/29     9/41
MN500 000627' XN      8/52    10/01
MT0   000000'         1/24     1/36
MT0A  000011- EN      1/02     1/18
MT0T  000012'         1/18     1/35
MT0.0 000004'         1/28     1/37
MT1   000002'         1/26     1/39
MT1A  000013- EN      1/02     1/19
MT1T  000015'         1/19     1/38
MT1.0 000007'         1/31     1/40
MTBHL 000531' XN      6/33     8/41
MTC   000000- EN      1/03     1/09
MTCCN 000004- EN      1/03     1/13     3/34     4/60     5/10     5/44     7/31
                      8/03     8/11     8/22     8/24
MTCS  000002- EN      1/03     1/11
MTEA3 000655'        10/25    10/33    10/34
MTEND 000644'         9/04    10/08    10/24    10/25
MTFLG 000006- EN      1/04     1/15     6/51     6/57     7/06
MTHL  000531'         7/27     7/37     8/41
MTN   000001- EN      1/03     1/10     7/17
MTNCN 000005- EN      1/03     1/14     3/35     4/28     4/56     5/19     5/59
                      7/59     8/23    10/07    10/15
MTNDV 000010- EN      1/02     1/17     3/44     4/21     4/24     4/53     5/12
                      5/38     8/14
MTNS  000003- EN      1/03     1/12     6/40     6/46     7/12     7/40
MTSW  000537'         7/54     8/47
MTTK1 000007- EN      1/04     1/16     8/39
MTWRI 000530'         7/35     8/40
MT.AS 000005$ XD      1/07     5/08     5/17     8/09    10/13
MT.DU 000006$ XD      1/07     3/42     4/15
MT.EF 000011$ XD      1/07    10/26
MT.ER 000015$ XD      7/50     7/57
MT.HD 000010$ XD      1/07     8/12     8/26
MT.RL 000012$ XD      1/07     4/37     4/50    10/31
MT.UN 000007$ XD      1/07     4/47
MUINI 000004$ XD      1/06     4/32
P.CR  000001$ XD      1/06     3/30     3/38     4/05     4/11     4/43     5/41
                      9/18
P.ER  000014$ XD      6/34    10/32
P.NL  000002$ XD      1/06     5/56     6/02     6/05
Q00   000020'         1/42     3/51
Q00B  000210'         3/29     3/51     4/04
Q01   000035'         1/55     3/52
Q01B  000211'         3/37     3/52     4/10
Q02   000047'         2/05     5/26     8/48
Q02B  000311'         5/07     5/26
Q02BB 000540'         8/08     8/48
Q03   000063'         2/17     5/27    10/23
Q03B  000643'        10/12    10/23
Q03BB 000312'         5/16     5/27
Q07   000100'         2/30     5/28
Q07B  000313'         4/42     5/28
Q09   000120'         2/46     9/33
Q09B  000602'         9/17     9/33
Q51   000123'         2/49     6/17
```

```
 0012 TR000

Q51B   000361'          5/37     6/17
Q52    000127'          2/53     6/18
 52B   000362'          5/40     6/18
Q53    000370'          5/52     6/19     6/24
Q53B1  000363'          5/48     6/19
Q54    000133'          2/57     6/20
Q54B   000364'          5/43     6/20
Q55    000142'          3/04     6/21
Q55B   000365'          5/58     6/21
Q56    000150'          3/10     6/22
Q56B   000366'          6/01     6/22
Q57    000160'          3/18     6/23
Q57B   000367'          6/04     6/23
Q90    000360'          5/49     6/16
R.LN   000003$ XD       1/06     3/39     4/12     4/44
S.PUS  000013$ XD       6/34    10/29
TK3ID  000577'          8/59     9/30
TK4ID  000600'          9/08     9/31
TK5AD  000627'          9/51    10/01
TK5ID  000601'          9/13     9/32
TKEND  000626'          9/50     9/60
TPOVF  000533' EN       6/32     7/25     8/43
TR000  000152' EN       1/01     3/20     3/21     3/24    10/35
TR001  000000' EN       1/02     1/22
TR010  000174'          3/37     3/40     3/43
TR0A0  000204'          3/24     3/47     3/59     5/21
TR0A1  000205'          3/25     3/48     3/60     5/22
TR0A2  000206'          3/26     3/49     4/01     5/23
  0A3  000207'          3/27     3/50     4/02     5/24
TR100  000214' EN       3/55     3/56     3/59
TR110  000224'          4/10     4/16     4/45
TR120  000236'          4/19     4/24
TR130  000251' EN       3/56     4/34     4/40
TR132  000251'          4/42     4/48
TR140  000260'          4/30     4/38     4/52
TR150  000262'          4/13     4/22     4/26     4/56
TR160  000272'          3/45     5/02     5/07
TR170  000276'          5/05     5/12
TR190  000305'          4/58     5/14     5/21
TR200  000377' EN       6/31     6/32     6/38     7/03     7/19     7/33     7/47
                        8/37
TR205  000422'          6/53     7/01
TR210  000430'          6/59     7/09
TR220  000451' EN       7/23     7/30     7/31
TR240  000467' EN       7/46     7/48     7/49     7/54
TR250  000513' EN       8/01     8/20     8/22
TR260  000522'          8/16     8/32
TR270  000523'          8/30     8/33
TR290  000526'          8/19     8/37
TR300  000542' EN       8/05     8/27     8/50     8/51     8/54
TR310  000553'          9/01     9/05
TR320  000561'          9/10     9/13
TR330  000564'          9/03     9/11     9/17
TR340  000566'          9/15     9/20
  350  000603' EN       8/18     8/51     9/36    10/17
TR360  000616'          9/43     9/50
TR390  000622'          9/48     9/55
TR3A0  000572'          8/54     9/20     9/25     9/39     9/55
```

```
  0013 TR000

TR3A1 000573'         8/55    9/21    9/26    9/38    9/56
TR3A2 000574'         8/56    9/22    9/27    9/37    9/57
TR3A3 000575'         8/57    9/23    9/28    9/36    9/58
TR400 000630' EN     10/02   10/03   10/07   10/22
TR410 000632'        10/12   10/21
TR4A1 000642'         8/29   10/22
TR4A2 000641'         9/32   10/21
TR4WD 000640'         9/33   10/20
TR500 000314' EN      4/08    5/04    5/29    5/30    5/32
TR520 000336'         5/46    5/54
TR530 000345'         5/55    6/02
TR540 000354'         5/32    6/07    6/12
TR5A1 000355'         5/33    6/08    6/13
TR5A2 000356'         5/34    6/09    6/14
TR5A3 000357'         5/35    6/10    6/15
VB0SB 000212'         3/31    3/53
VB2SB 000213'         3/54    4/06
VB800 000212' XN      1/05    3/53
VB820 000213' XN      1/05    3/54
.ABOR 000556' XN      8/52    8/60    9/09
.KILL 000637' XN      1/05   10/18
.REC  000400' XN      1/05    6/39
.TASK 000620' XN      1/05    8/34    9/52
.TIDK 000562' XN      1/05    9/14
.TIDR 000613' XN      8/52    9/46
```

```
0001 MT000 MACRO REV 06.50                         20:40:08 10/31/80
                                  .TITL   MT000
                                  .ENT    MT000
                                  .EXTD   P.CR,P.NL,R.LN,GT.CH,IBUF,DIG.,OC.AS
                                  .EXTD   MTCCN,MTNCN,MT0A,MT1A,CK.CR
                                  .NREL
             000001               .TXTM   1
                          MT000:
   000000'020124 Q04:     .TXT    / TYPE UNIT # (0 OR 1): /
             054520
             042440
             052516
             044524
             020043
             020050
             030040
             047522
             020061
             024472
             020000
   000014'044514 Q05:     .TXT    /ILLEGAL UNIT,/
             046105
             043501
             046040
             052516
             044524
             026000
   000023'052516 Q06:     .TXT    /UNIT IN USE,/
             044524
             020111
             047040
             052523
             042454
             000000

                          .TITL   MTASG   ;ASSIGN,INIT & OPEN
                          .ENT    MTASG,MT.AS
                    ;
                    ;  SUBROUTINE TO ASSIGN A TAPE DRIVE, INIT & OPEN IT
                    ;     AC0 - INPUT BYTE POINTER TO MESSAGE
                    ;     AC2 - RETURN POINTER TO MT TABLE
                    ;     RETURN+1 - USER RESPONSE = JUST CR
                    ;     RETURN -OK
                    ;
                          .ZREL
   000000-000032'MT.AS:   MTASG
                          .NREL
   000032'040432 MTASG:   STA     0,MTAA0
   000033'044432          STA     1,MTAA1           ;SAVE REGISTERS
   000034'054433          STA     3,MTAA3
                    ;TYPE PROMPTING, GET UNIT
   000035'006001$MTA05:   JSR     @P.CR             ;TYPE USER MESSAGE
   000036'020432 MTA10:   LDA     0,Q04B
   000037'006002$         JSR     @P.NL             ;TYPE PROMPT
   000040'006003$         JSR     @R.LN             ;READ RESPONSE
   000041'000420          JMP     MTA92             ;ERROR RETURN
   000042'006002-         JSR     @MT.UN            ;GET UNIT NUMBER
   000043'000773          JMP     MTA10             ;ILLEGAL - TRY AGAIN
                    ;CHECK UNIT UNUSED
   000044'024010$         LDA     1,MTCCN
   000045'146405          SUB     2,1,SNR
   000046'000407          JMP     MTA20             ;ERROR
```

```
 0002 MT000
                   ;
000047'024011$        LDA      1,MTNCN
000050'146405         SUB      2,1,SNR
000051'000404         JMP      MTA20             ;ERROR
                   ;    OK
000052'050414         STA      2,MTAA2           ;SAVE FOR RETURN
000053'006006-        JSR      @MT.IN            ;INIT AND OPEN
000054'000403         JMP      MTA90
                   ;    UNIT IN USE
000055'020414 MTA20:  LDA      0,Q06B
000056'000757         JMP      MTA05
                   ;RETURN
000057'010410 MTA90:  ISZ      MTAA3             ;RETURN+1 - OK
000060'030406         LDA      2,MTAA2
000061'024404 MTA92:  LDA      1,MTAA1
000062'020402         LDA      0,MTAA0
000063'002404         JMP      @MTAA3
                   ;
000064'000000 MTAA0:  .-.
000065'000000 MTAA1:  .-.
000066'000000 MTAA2:  .-.
000067'000000 MTAA3:  .-.
                   ;
000070'000000"Q04B:   Q04*2
000071'000046"Q06B:   Q06*2
                      .TITL    MTDV     ;# DEVICES
                      .ENT     MTDV,MT.DV
                   ;
                   ;    ROUTINE TO DECODE # TAPE DEVICES
                   ;        AC1 - RETURN 1 OR 2
                   ;        RETURN - NEITHER 1 NOR 2; FORMAT ERROR
                   ;        RETURN+1 - OK
                   ;
                      .ZREL
000001-000072'MT.DV:  MTDV
                      .NREL
000072'054416 MTDV:   STA      3,MTVA3
000073'020005$        LDA      0,IBUF
000074'006006$        JSR      @DIG.             ;GET NUMBER
000075'151005         MOV      2,2,SNR           ;CHECK # DIGITS
000076'000411         JMP      MTV90
000077'125005         MOV      1,1,SNR           ;CHECK FOR 0
000100'000407         JMP      MTV90
000101'030410         LDA      2,K2
000102'132433         SUBZ#    1,2,SNC
000103'000404         JMP      MTV90             ;BAD
000104'006014$        JSR      @CK.CR
000105'000402         JMP      MTV90             ;NO CR
                   ;RETURN
000106'010402         ISZ      MTVA3             ;OK - RETURN+1
000107'002401 MTV90:  JMP      @MTVA3
                   ;
000110'000000 MTVA3:  .-.
000111'000002 K2:     2
                      .TITL    MTUN     ;TAPE UNIT
                      .ENT     MTUN,MT.UN
                   ;
                   ;    ROUTINE TO DECODE TAPE UNIT #
                   ;        AC2 - RETURNS MT0A OR MT1A
                   ;        AC1 - RETURNS 0 OR 1
```

```
                ;    RETURN - NEITHER 0 NOR 1
                ;    RETURN+1 - OK
                ;
                     .ZREL
000002-000112'MT.UN: MTUN
                     .NREL
000112'054423 MTUN:  STA     3,MTUA3
000113'020005$       LDA     0,IBUF
000114'006006$       JSR     @DIG.               ;GET NUMBER
000115'151005        MOV     2,2,SNR
000116'000405        JMP     MTU10
                ;
000117'125005        MOV     1,1,SNR             ;CHECK FOR 0
000120'000406        JMP     MTU20
                ;
000121'125235        MOVZR#  1,1.SNR             ;CHECK FOR 1
000122'000406        JMP     MTU30
                ;
000123'020413 MTU10: LDA     0,Q05B              ;NEITHER - TYPE ERROR
000124'006001$       JSR     @P.CR
000125'000407        JMP     MTU92
                ;UNIT 0
000126'030012$MTU20: LDA     2,MT0A
000127'000402        JMP     MTU40
                ;UNIT 1
000130'030013$MTU30: LDA     2,MT1A
                ;CHECK CR
000131'006014$MTU40: JSR     @CK.CR
000132'000402        JMP     MTU92               ;NO CR
                ;RETURN
000133'010402        ISZ     MTUA3
000134'002401 MTU92: JMP     @MTUA3
                ;
000135'000000 MTUA3: .-.
000136'000030"Q05B:  Q05*2
                     .TITL   MTHED   ;HEADER
                     .ENT    MTHED,MT.HD
                     .EXTD   LGMNS,MTFLG,HIADR
                     .EXTN   MTHDR,MTHLN
                ;
                ;  SUBROUTINE TO BUILD & WRITE HEADER TO TAPE FILE
                ;    AC2 - ADR TAPE BLOCK
                ;      -
                     .ZREL
000003-000137'MT.HD: MTHED
                     .NREL
000137'040442 MTHED: STA     0,MTHA0             ;SAVE REGISTERS
000140'044442        STA     1,MTHA1
000141'050442        STA     2,MTHA2
000142'054442        STA     3,MTHA3
                ;BUILD TAPE HEADER
                ;  DATE & TIME
000143'006017        .SYSTEM
000144'021003        .GTOD
000145'000000        .-.
000146'034440        LDA     3,MTHAD
000147'041405        STA     0,5,3
000150'045404        STA     1,4,3
000151'051403        STA     2,3,3
000152'006017        .SYSTEM
```

```
000153'021006        .GDAY
000154'000000        .-.
000155'034431        LDA     3,MTHAD
000156'041402        STA     0,2,3
000157'045401        STA     1,1,3
000160'051400        STA     2,0,3
                   ;    ANY OTHER HEADER INFO
000161'020015$       LDA     0,LGMNS           ;LOGGING INTERVAL
000162'041406        STA     0,6,3
000163'020016$       LDA     0,MTFLG           ;RECORDING MODE
000164'041407        STA     0,7,3
000165'020017$       LDA     0,HIADR           ;TOP ADR
000166'041410        STA     0,10,3
                   ;WRITE TAPE HEADER
000167'032414        LDA     2,@MTHA2          ;GET CHANNEL
000170'024415        LDA     1,MTHWC           ;GET COMMAND/COUNT
000171'020415        LDA     0,MTHAD           ;GET ADDRES
000172'006017        .SYSTEM
000173'023077        .MTDIO  77
000174'004413        JSR     MTDER
                   ;RETURN
000175'020404        LDA     0,MTHA0
000176'024404        LDA     1,MTHA1
000177'030404        LDA     2,MTHA2
000200'002404        JMP     @MTHA3
                   ;
000201'000000 MTHA0: .-.
000202'000000 MTHA1: .-.
000203'000000 MTHA2: .-.
000204'000000 MTHA3: .-.
000205'177777 MTHWC: MTHLN
000206'177777 MTHAD: MTHDR
                     .TITL   MTDER
                     .ENT    MTDER,MT.ER
                     .ZREL
000004-000207'MT.ER: MTDER
                     .NREL
000207'040424 MTDER: STA     0,MTDA0
000210'044424        STA     1,MTDA1
000211'050424        STA     2,MTDA2
000212'054424        STA     3,MTDA3
    -              ;TYPE "WRITE"
000213'020424        LDA     0,Q23B
000214'006001$       JSR     @P.CR
                   ;CHECK RDOS ERROR OR STATUS
000215'151132        MOVZL#  2,2,SZC
000216'000404        JMP     MTD30
000217'004572        JSR     MTXER
000220'000401        JMP     .+1
000221'000406        JMP     MTD90
                   ;STATUS WORD RETURNED
000222'020417 MTD30: LDA     0,Q34BX
000223'145000        MOV     2,1
000224'006007$       JSR     @OC.AS
000225'020413        LDA     0,Q34B
000226'006002$       JSR     @P.NL
                   ;RETURN
000227'020404 MTD90: LDA     0,MTDA0
000230'024404        LDA     1,MTDA1
000231'030404        LDA     2,MTDA2
```

```
000232'002404          JMP     @MTDA3
                  ;
000233'000000 MTDA0:   .-.
000234'000000 MTDA1:   .-.
000235'000000 MTDA2:   .-.
000236'000000 MTDA3:   .-.
000237'001336"Q23B:    Q23*2
000240'001565"Q34B:    Q34*2
000241'001503"Q34BX:   Q34*2+15
                       .TITL   MTEF    ;2 EOF'S
                       .ENT    MTEF,MT.EF
                  ;
                  ;    SUBROUTINE TO WRITE TWO EOF'S TO TAPE FILE
                  ;        AC2 - ADR TAPE BLOCK
                  ;
                       .ZREL
000005-000242'MT.EF:   MTEF
                       .NREL
000242'040422 MTEF:    STA     0,MTEA0
000243'044422          STA     1,MTEA1
000244'050422          STA     2,MTEA2
000245'054422          STA     3,MTEA3
                  ;INDICATE  EOT  WRITTEN
000246'020423          LDA     0,JPOS
000247'006001$         JSR     @P.CR
                  ;WRITE EOF
000250'024420          LDA     1,MTEOF
000251'032415          LDA     2,@MTEA2
000252'006017          .SYSTEM
000253'023077          .MTDIO  77
                  ;WRITE EOF
000254'024414          LDA     1,MTEOF
000255'032411          LDA     2,@MTEA2
000256'006017          .SYSTEM
000257'023077          .MTDIO  77
                  ;RETURN
000260'020404          LDA     0,MTEA0
000261'024404          LDA     1,MTEA1
000262'030404          LDA     2,MTEA2
000263'002404          JMP     @MTEA3
                  ;
000264'000000 MTEA0:   .-.
000265'000000 MTEA1:   .-.
000266'000000 MTEA2:   .-.
000267'000000 MTEA3:   .-.
000270'060000 MTEOF:   6B3
000271'001606"JPOS:    Q35*2
                       .TITL   MTINI   ;INIT/OPEN
                       .ENT    MTINI,MT.IN
                  ;
                  ;    SUBROUTINE TO INITIALIZE AND OPEN MAG TAPE
                  ;        AC2 - ADR TAPE BLOCK
                  ;
                       .ZREL
000006-000272'MT.IN:   MTINI
                       .NREL
000272'040502 MTINI:   STA     0,MTIA0
000273'044502          STA     1,MTIA1
000274'050502          STA     2,MTIA2
000275'054502          STA     3,MTIA3
```

```
                         ;INITIALIZE
   000276'102400               SUB     0,0
   000277'040552               STA     0,MTXCT
                         ;
   000300'021001 MTI10:  LDA     0,1,2               ;BYTE POINTER UNIT
   000301'125000               ADC     1,1                 ;FULL INIT
   000302'006017               .SYSTEM
   000303'004000               .INIT
   000304'000402               JMP     MTI20
   000305'000431               JMP     MTI50               ;OK
                         ;    INIT ERROR
   000306'020475 MTI20:  LDA     0,Q21B
   000307'006001$              JSR     @P.CR
                         ;
   000310'024470               LDA     1,K10
   000311'146404               SUB     2,1,SZR             ;CHECK WRITE PROTECT
   000312'000404               JMP     MTI30
   000313'020474               LDA     0,Q32B              ;IF YES,TYPE MESSAGE
   000314'006002$              JSR     @P.NL
   000315'000413               JMP     MTI45               ;GO GET OPERATOR FIX
                         ;
   000316'024464 MTI30:  LDA     1,K45               ;CHECK ALREADY INIT
   000317'146404               SUB     2,1,SZR
   000320'000406               JMP     MTI40
   000321'020464               LDA     0,Q29B              ;IF YES, TYPE MESSAGE
   000322'006002$              JSR     @P.NL
   000323'030453               LDA     2,MTIA2
   000324'004537               JSR     MTREL               ;FIX BY RELEASING
   000325'000753               JMP     MTI10               ;TRY AGAIN
                         ;
   000326'004463 MTI40:  JSR     MTXER               ;CHECK OTHER ERRORS
   000327'000751               JMP     MTI10               ;UNIT NOT READY
                         ;
   000330'020460 MTI45:  LDA     0,Q33B              ;TYPE "FIX, TYPE CR"
   000331'006001$              JSR     @P.CR
   000332'006003$              JSR     @R.LN
   000333'000401               JMP     .+1
   000334'030442               LDA     2,MTIA2
   000335'000743               JMP     MTI10               ;TRY AGAIN
                         ;OPEN
   000336'102400 MTI50:  SUB     0,0
   000337'040512               STA     0,MTXCT
                         ;
   000340'030436               LDA     2,MTIA2
   000341'021002               LDA     0,2,2               ;BYTE POINTER FILE
   000342'031000               LDA     2,0,2               ;CHANNEL #
   000343'006017               .SYSTEM
   000344'025077               .MTOPD  77
   000345'000402               JMP     MTI60
   000346'000422               JMP     MTI90
                         ;    OPEN ERROR
   000347'020435 MTI60:  LDA     0,Q22B
   000350'006001$              JSR     @P.CR
                         ;
   000351'024430               LDA     1,K21               ;CHECK CHANNEL IN USE
   000352'146404               SUB     2,1,SZR
   000353'000406               JMP     MTI70
   000354'020432               LDA     0,Q30B              ;IF YES, TYPE MESSAGE
   000355'006002$              JSR     @P.NL
   000356'030420               LDA     2,MTIA2
```

```
 0007 MT000
 000357'004504          JSR      MTREL            ;FIX BY CLOSING
 000360'000720          JMP      MTI10
                 ;
 000361'004430 MTI70:   JSR      MTXER            ;CHECK OTHER ERRORS
 000362'000754          JMP      MTI50            ;UNIT NOT READY
                 ;
 000363'020425          LDA      0,Q33B
 000364'006001$         JSR      @P.CR
 000365'006003$         JSR      @R.LN
 000366'000401          JMP      .+1
 000367'000747          JMP      MTI50
                 ;RETURN
 000370'020404 MTI90:   LDA      0,MTIA0
 000371'024404          LDA      1,MTIA1
 000372'030404          LDA      2,MTIA2
 000373'002404          JMP      @MTIA3
                 ;
 000374'000000 MTIA0:   .-.
 000375'000000 MTIA1:   .-.
 000376'000000 MTIA2:   .-.
 000377'000000 MTIA3:   .-.
 000400'000010 K10:     10
 000401'000021 K21:     21
 000402'000045 K45:     45
 000403'001316"Q21B:    Q21*2
 000404'001326"Q22B:    Q22*2
 000405'001450"Q29B:    Q29*2
 000406'001466"Q30B:    Q30*2
 000407'001526"Q32B:    Q32*2
 000410'001546"Q33B:    Q33*2
                        .TITL    MTXER   ;COMMON ERRORS
                        .EXTN    MTADC
 000411'054435 MTXER:   STA      3,MTXA3
 000412'044435          STA      1,MTXA1
 000413'040435          STA      0,MTXA0
                 ;
 000414'024440          LDA      1,K31            ;CHECK UNIT NOT READY
 000415'146404          SUB      2,1,SZR
 000416'000412          JMP      MTX10
 000417'020441          LDA      0,Q27B           ;IF YES, TYPE MESSAGE
 000420'024431          LDA      1,MTXCT
 000421'125004          MOV      1,1,SZR
 000422'020437          LDA      0,Q27B.          ; OR .
 000423'010426          ISZ      MTXCT
                 ;
 000424'034426          LDA      3,AMTAD
 000425'024426          LDA      1,K4
 000426'045412          STA      1,12,3           ;RESET WAIT COUNT
 000427'000413          JMP      MTX70
                 ;
 000430'010416 MTX10:   ISZ      MTXA3
 000431'024424          LDA      1,K36
 000432'146404          SUB      2,1,SZR
 000433'000403          JMP      MTX60
 000434'020426          LDA      0,Q28B
 000435'000405          JMP      MTX70
                 ;
 000436'020421 MTX60:   LDA      0,Q26B6
 000437'145000          MOV      2,1
 000440'006007$         JSR      @DC.AS
```

```
000441'020415             LDA     0,026B

000442'006002$MTX70:      JSR     @P.NL
000443'020405             LDA     0,MTXA0
000444'024403             LDA     1,MTXA1
000445'002401             JMP     @MTXA3

000446'000000 MTXA3:      .-.
000447'000000 MTXA1:      .-.
000450'000000 MTXA0:      .-.
000451'000000 MTXCT:      .-.
000452'177777 AMTAD:      MTADC
000453'000004 K4:         4
000454'000031 K31:        31
000455'000036 K36:        36
000456'001366"Q26B:       Q26*2
000457'001374"Q26B6:      Q26*2+6
000460'001404"Q27B:       Q27*2
000461'001422"Q27B.:      Q27*2+16
000462'001424"Q28B:       Q28*2
                          .TITL   MTREL    ;RELEASE, CLOSE, DEACTIVATE
                          .ENT    MTREL,MT.RL
                   ;
                   ;    SUBROUTINE TO CLOSE & RELEASE MAG TAPE
                   ;        AC2 - ADDR TAPE BLOCK
                   ;
                          .ZREL
000007-000463'MT.RL:      MTREL
                          .NREL
000463'040454 MTREL:      STA     0,MTRA0
000464'044454             STA     1,MTRA1
000465'050454             STA     2,MTRA2
000466'054454             STA     3,MTRA3
                   ;CLOSE CHANNEL
000467'031000             LDA     2,0,2
000470'006017             .SYSTEM
000471'014477             .CLOSE  77
000472'000402             JMP     MTR10
000473'000415             JMP     MTR40
                   ;    CLOSE ERROR
000474'020450 MTR10:      LDA     0,Q24B
000475'006001$            JSR     @P.CR
                   ;
000476'024445             LDA     1,K15              ;CHECK "NOT OPEN"
000477'146404             SUB     2,1,SZR
000500'000404             JMP     MTR20
000501'020445             LDA     0,Q31B             ;IF YES,TYPE MESSAGE
000502'006002$            JSR     @P.NL
000503'000403             JMP     MTR30              ;GO ON
                   ;
000504'004705 MTR20:      JSR     MTXER              ;CHECK OTHER ERRORS
000505'000401             JMP     MTR30              ;GO ON
                   ;RELEASE MTU
000506'102400 MTR30:      SUB     0,0
000507'040742             STA     0,MTXCT
                   ;
000510'030431 MTR40:      LDA     2,MTRA2
000511'021001             LDA     0,1,2              ;BYTE POINTER TO UNIT
000512'006017             .SYSTEM
000513'002400             .RLSE
```

```
0009 MT000
000514'000402          JMP     MTR50
000515'000405          JMP     MTR80
000516'020427 MTR50:   LDA     0,Q25B
000517'006001$         JSR     @P.CR
000520'004571          JSR     MTXER
000521'000767          JMP     MTR40
              ;
              ;DE-ACTIVATE CHANNEL
000522'030417 MTR80:   LDA     2,MTRA2                ;GET POINTER
000523'024011$         LDA     1,MTNCN                ;CHECK NON-CURR
000524'146404          SUB     2,1,SZR
000525'024011$         LDA     1,MTNCN
000526'044011$         STA     1,MTNCN
000527'020010$         LDA     0,MTCCN                ;CHECK CURR
000530'142404          SUB     2,0,SZR
000531'000403          JMP     MTR90
000532'044010$         STA     1,MTCCN                ;FLIP IF
000533'040011$         STA     0,MTNCN                ; NECESSARY
              ;RETURN
000534'020403 MTR90:   LDA     0,MTRA0
000535'024403          LDA     1,MTRA1
000536'002404          JMP     @MTRA3
              ;
000537'000000 MTRA0:   .-.
000540'000000 MTRA1:   .-.
000541'000000 MTRA2:   .-.
000542'000000 MTRA3:   .-.
000543'000015 K15:     15
000544'001346"Q24B:    Q24*2
000545'001356"Q25B:    Q25*2
000546'001504"Q31B:    Q31*2
                       .TITL   MT000
000547'044516 Q21:     .TXT    /INIT: /
       044524
       035040
       000000
000553'047520 Q22:     .TXT    /OPEN: /
       042516
       035040
       000000
000557'053522 Q23:     .TXT    /WRITE: /
       044524
       042472
       020000
000563'041514 Q24:     .TXT    /CLOSE: /
       047523
       042472
       020000
000567'051105 Q25:     .TXT    /RELE: /
       046105
       035040
       000000
000573'042522 Q26:     .TXT    /ERROR XXXXXX<15>/
       051117
       051040
       054130
       054130
       054130
       006400
000602'052516 Q27:     .TXT    /UNIT NOT READY./
```

```
0010 MT000
                044524
                020116
                047524
                020122
                042501
                042131
                027000
        000612'052516 Q28:    .TXT    /UNIT NOT IN SYSTEM<15>/
                044524
                020116
                047524
                020111
                047040
                051531
                051524
                042515
                006400
        000624'040514 Q29:    .TXT    /ALREADY INIT<15>/
                051105
                040504
                054440
                044516
                044524
                006400
        000633'041510 Q30:    .TXT    /CHAN IN USE<15>/
                040516
                020111
                047040
                052523
                042415
                000000
        000642'041510 Q31:    .TXT    /CHAN NOT IN USE<15>/
                040516
                020116
                047524
                020111
                047040
                052523
                042415
                000000
        000653'053522 Q32:    .TXT    /WRITE PROTECT: /
                044524
                042440
                050122
                047524
                042503
                052072
                020000
        000663'043111 Q33:    .TXT    /FIX, TYPE CR: /
                054054
                020124
                054520
                042440
                041522
                035040
                000000
        000673'051524 Q34:    .TXT    /STATUS XXXXXX<15>/
                040524
                052523
                020130
```

```
 0011 MT000
01          054130
02          054130
03          054015
04          000000
05 000703'005040 035:    .TXT    /<12> EOT WRITTEN<15>/
06          042517
07          052040
08          053522
09          044524
10          052105
11          047015
12          000000
13                       .END

**00000 TOTAL ERRORS, 00000 PASS 1 ERRORS
```

```
   0012 MT000

  AMTAD 000452'          7/46     8/12
  CK.CR 000014$ XD       1/04     2/47     3/29
  DIG.  000006$ XD       1/03     2/39     3/09
  GT.CH 000004$ XD       1/03
  HIADR 000017$ XD       3/38     4/12
  IBUF  000005$ XD       1/03     2/38     3/08
  JPOS  000271'          5/24     5/47
  K10   000400'          6/15     7/22
  K15   000543'          8/44     9/28
  K2    000111'          2/44     2/54
  K21   000401'          6/55     7/23
  K31   000454'          7/37     8/14
  K36   000455'          7/52     8/15
  K4    000453'          7/47     8/13
  K45   000402'          6/22     7/24
  LCMNS 000015$ XD       3/38     4/08
  MT000 000000' EN       1/01     1/02     1/07     9/32
  MT0A  000012$ XD       1/04     3/23
  MT1A  000013$ XD       1/04     3/26
  MTA05 000035'          1/50     2/11
  MTA10 000036'          1/51     1/56
  MTA20 000055'          1/60     2/04     2/10
  MTA90 000057'          2/08     2/13
  MTA92 000061'          1/54     2/15
  MTAA0 000064'          1/46     2/16     2/19
  MTAA1 000065'          1/47     2/15     2/20
  MTAA2 000066'          2/06     2/14     2/21
  MTAA3 000067'          1/48     2/13     2/17     2/22
  MTADC 000452' XN       7/32     8/12
  MTASG 000032' EN       1/34     1/35     1/44     1/46
  MTCCN 000010$ XD       1/04     1/58     9/14     9/17
  MTD30 000222'          4/47     4/52
  MTD90 000227'          4/50     4/58
  MTDA0 000233'          4/38     4/58     5/03
  MTDA1 000234'          4/39     4/59     5/04
  MTDA2 000235'          4/40     4/60     5/05
  MTDA3 000236'          4/41     5/01     5/06
  MTDER 000207' EN       4/20     4/33     4/34     4/36     4/38
  MTDV  000072' EN       2/26     2/27     2/35     2/37
  MTEA0 000264'          5/19     5/37     5/42
  MTEA1 000265'          5/20     5/38     5/43
  MTEA2 000266'          5/21     5/28     5/33     5/39     5/44
  MTEA3 000267'          5/22     5/40     5/45
  MTEF  000242' EN       5/10     5/11     5/17     5/13
  MTEOF 000270'          5/27     5/32     5/46
  MTFLC 000016$ XD       3/38     4/10
  MTHA0 000201'          3/47     4/22     4/27
  MTHA1 000202'          3/48     4/23     4/28
  MTHA2 000203'          3/49     4/15     4/24     4/29
  MTHA3 000204'          3/50     4/25     4/30
  MTHAD 000206'          3/56     4/03     4/17     4/32
  MTHDR 000206' XN       3/39     4/32
  MTHED 000137' EN       3/36     3/37     3/45     3/47
  MTHLN 000205' XN       3/39     4/31
  MTHWC 000205'          4/16     4/31
  MTI10 000300'          6/05     6/29     6/32     6/39     7/02
  MTI20 000306'          6/09     6/13
  MTI30 000316'          6/17     6/22
  MTI40 000326'          6/24     6/31
  0013 MT000

MTI45  000330'      6/20   6/34
MTI50  000336'      6/10   6/41   7/05   7/11
MTI60  000347'      6/49   6/52
MTI70  000361'      6/57   7/04
MTI90  000370'      6/50   7/13
MTIA0  000374'      5/57   7/13   7/18
MTIA1  000375'      5/58   7/14   7/19
MTIA2  000376'      5/59   6/27   6/38   6/44   6/50   7/15   7/20
MTIA3  000377'      5/60   7/16   7/21
MTINI  000272' EN   5/48   5/49   5/55   5/57
MTNCN  000011$ XD   1/04   2/02   9/10   9/12   9/13   9/18
MTR10  000474'      8/38   8/41
MTR20  000504'      8/46   8/51
MTR30  000506'      8/49   8/52   8/54
MTR40  000510'      8/39   8/57   9/06
MTR50  000516'      9/01   9/03
MTR80  000522'      9/02   9/09
MTR90  000534'      9/16   9/20
MTRA0  000537'      8/30   9/20   9/24
MTRA1  000540'      8/31   9/21   9/25
MTRA2  000541'      8/32   8/57   9/09   9/26
MTRA3  000542'      8/33   9/22   9/27
MTREL  000463' EN   6/28   7/01   8/21   8/22   8/29   8/30
MTU10  000123'      3/11   3/19
MTU20  000126'      3/14   3/23
MTU30  000130'      3/17   3/26
MTU40  000131'      3/24   3/28
MTU92  000134'      3/21   3/29   3/32
MTUA3  000135'      3/07   3/31   3/32   3/34
MTUN   000112' EN   2/55   2/56   3/05   3/07
MTV90  000107'      2/41   2/43   2/46   2/48   2/51
MTVA3  000110'      2/37   2/50   2/51   2/53
MTX10  000430'      7/39   7/51
MTX60  000436'      7/54   7/58
MTX70  000442'      7/49   7/56   8/03
MTXA0  000450'      7/35   8/04   8/10
MTXA1  000447'      7/34   8/05   8/09
MTXA3  000446'      7/33   7/51   8/06   8/08
MTXCT  000451'      6/03   6/42   7/41   7/44   8/11   8/55
MTXER  000411'      4/48   6/31   7/04   7/31   7/33   8/51   9/05
MT.AS  000030- EN   1/35   1/44
MT.DV  000001- EN   2/27   2/35
MT.EF  000005- EN   5/11   5/17
MT.ER  000004- EN   4/34   4/36
MT.HD  000003- EN   3/37   3/45
MT.IN  000006- EN   2/07   5/49   5/55
MT.RL  000007- EN   8/22   8/28
MT.UN  000002- EN   1/55   2/56   3/05
OC.AS  000007$ XD   1/03   4/54   7/60
P.CR   000001$ XD   1/03   1/50   3/20   4/44   5/25   6/13   6/35
                    6/53   7/09   8/42   9/04
P.NL   000002$ XD   1/03   1/52   4/56   6/19   6/26   6/59   9/07
                    8/48
004    000000'      1/08   2/24
0048   000070'      1/51   2/24
005    000014'      1/20   3/35
005B   000136'      3/19   3/35
006    000023'      1/27   2/25
```

```
  0014 MT000

006B   000071'          2/10    2/25
021    000547'          7/25    9/33
021B   000403'          6/12    7/25
022    000553'          7/26    9/37
022B   000404'          6/52    7/26
023    000557'          5/07    9/41
023B   000237'          4/43    5/07
024    000563'          9/29    9/45
024B   000544'          8/41    9/29
025    000567'          9/30    9/49
025B   000545'          9/03    9/30
026    000573'          8/16    8/17     9/53
026B   000456'          8/01    8/16
026BS  000457'          7/58    8/17
027    000602'          8/18    8/19     9/60
027B   000460'          7/40    8/18
027B.  000461'          7/43    8/19
028    000612'          8/20   10/08
028B   000462'          7/55    8/20
029    000624'          7/27   10/18
029B   000405'          6/25    7/27
030    000633'          7/28   10/25
030B   000406'          6/58    7/28
031    000642'          9/31   10/32
031B   000546'          8/47    9/31
032    000653'          7/29   10/41
032B   00C407'          6/18    7/29
033    000663'          7/30   10/49
033B   000410'          6/34    7/07     7/30
034    000673'          5/08    5/09    10/57
034B   000240'          4/55    5/08
034BX  000241'          4/52    5/09
035    000703'          5/47   11/05
R.LN   000003$ XD       1/03    1/53     6/36     7/09
R
```

```
TYPE MVVB
        .TITL   VB000   ;VERBOSE VERSION
.
        .ENT    VB000,VB020,VB040,VB060,VB100,VB200,VB300
        .ENT    VB400,VB500,VB600,VB700,VB800,VB820,VB900
        .ENT    VB001,VB002,VBFLG,VB010,VB030,VB220,VB240,VB320
.
        .EXTD   P.CR,P.NL,R.LN,GT.CH,IBUF
.
        .TXTM   1
        .NREL
        .TITL   VB001   ;TEXT
VB001:
.
.    INTRODUCTION
.
C00     .TXT    ]<12> * * * WELCOME TO THE MV ARCTIC DATA
 GATHERING SYSTEM * * *<15>]
C02:    .TXT    ]<12>TYPE "H" FOLLOWED BY THE "RETURN" KE
Y FOR HELPFUL PRINTOUT.<15>]
C03:    .TXT    ]TYPE JUST THE "RETURN" KEY FOR MINIMAL P
ROMPTING: ]
.
.       GENERAL SYSTEM DESCRIPTION
.
A00:    .TXT    ]<12>SYSTEM STARTUP REQUIRES OPERATOR SEL
ECTION OF VARIOUS OPTIONS<15>]
A01:    .TXT    ]AND PARAMETERS.<15>]
A015:   .TXT    ] *A PROMPT CHARACTER (: OR ?) APPEARS WH
EN THE COMPUTER IS<15>]
A017    .TXT    ]  WAITING FOR INPUT FROM THE OPERATOR.<15>]
A03:    .TXT    ] *ONLY A SINGLE LETTER IS REQUIRED TO SE
LECT AN OPTION.  FOR<15>]
A04:    .TXT    ]  EXAMPLE, H WAS TYPED TO SELECT HELPFUL
 PRINTOUT.<15>]
A02:    .TXT    ] *EVERY RESPONSE TYPED MUST BE FOLLOWED
BY THE "RETURN" KEY<15>]
A025:   .TXT    ]  (ABBREVIATED "CR").<15>]
A05:    .TXT    ] *TYPE JUST "CR" TO PROCEED TO THE NEXT
SECTION.<12><15>]
A06:    .TXT    ]NO COMPUTER ACTION IS TAKEN UNTIL "CR" I
S TYPED.  THEREFORE, A<15>]
A07:    .TXT    ]TYPING ERROR MAY BE CORRECTED (AS DESCRI
BED BELOW) AT ANY TIME<15>]
A08:    .TXT    ]BEFORE "CR" IS TYPED.  AFTER "CR" IS TYP
ED, THE COMPUTER MUST<15>]
A09:    .TXT    ]RESPOND BEFORE FURTHER ENTRIES CAN BE MA
DE.<15>]
A11:    .TXT    ]<12>BEFORE "CR" IS TYPED, CORRECTION MAY
 BE ACCOMPLISHED IN TWO WAYS: <15>]
A12:    .TXT    ]BY DELETING THE LINE AND STARTING AGAIN
OR BY BACKSPACING ONE<15>]
A13:    .TXT    ]CHARACTER AT A TIME AND RE-ENTERING FROM
 THE ERROR.<15>]
A14:    .TXT    ] *TO DELETE A WHOLE LINE, TYPE THE BACKW
ARDS SLASH KEY (SHIFT "L").<15>]
A15:    .TXT    ]  THE BACK SLASH IS PRINTED AND A "CR" D
ONE AUTOMATICALLY.  START<15>]
A16:    .TXT    ]  TYPING AT THE BEGINNING OF THE LINE.<15>]
A17:    .TXT    ] *TO DELETE A FEW CHARACTERS, USE THE "R
```

```
UBOUT" KEY.  EACH TIME "RUBOUT"<15>]
A18      .TXT     ]  IS TYPED, ONE CHARACTER IS DELETED FRO
M THE END OF THE LINE.  WHEN<15>]
A19      .TXT     ]  ALL ERRONEOUS CHARACTERS HAVE BEEN DEL
ETED. START TYPING FROM THAT<15>]
A20      .TXT     ]  POINT.<15>]
.
.        DATE & TIME
.
A30      .TXT     ]<12>ENTER THE DATE WITH YEAR FIRST.  YEA
R SHOULD BE TWO DIGITS.<15>]
A31:     .TXT     ]MONTH AND DAY MAY BE ONE OR TWO DIGITS.
 ENTER THE TIME USING A<15>]
A32:     .TXT     ]24-HOUR CLOCK.  HOURS AND MINUTES MUST B
E ENTERED, SECONDS WILL<15>]
A33:     .TXT     ]BE SET TO ZERO IF OMITTED.  THE TIME IS
SET WHEN "CR" IS TYPED.<15>]
A34:     .TXT     ]OR, TYPE JUST "CR" TO LEAVE THE DATE OR
TIME INTACT.<12><15>]
.
.        SAMPLING PATTERN SELECTION
.
E00:     .TXT     /<12>THE SAMPLING PATTERN SPECIFIES WHIC
H SENSORS ARE SAMPLED, AT<15>/
E01:     .TXT     /WHAT RATES, AND IN WHICH MODES.<15>/
D00:     .TXT     ]<12>%INITIALIZE SETS ALL SENSORS TO DEFAULT
SAMPLING (RATE 2, PEAK<15>]
D01:     .TXT     ] MODE), BUT DOES NOT AFFECT THE EXISTING
 ALARM LIMITS.  SENSOR<15>]
D02:     .TXT     ] SETTINGS MAY BE EXAMINED OR MODIFIED.<15>]
D03:     .TXT     ]%MODIFY LEAVES THE EXISTING SAMPLING PAT
TERN AND ALARM LIMITS<15>]
D04:     .TXT     ] INTACT.  SENSOR SETTINGS MAY BE EXAMINE
D OR MODIFIED.<15>]
D05:     .TXT     ]%CHECK LEAVES THE EXISTING SAMPLING PATT
ERN AND ALARM LIMITS<15>]
D06:     .TXT     ] INTACT.  SENSOR SETTINGS MAY BE EXAMINE
D, BUT NOT MODIFIED.<12><15>]
D07:     .TXT     ]REMEMBER TO TYPE "CR" AFTER EVERY RESPON
SE.<15>]
D11:     .TXT     ]THE SAMPLING PATTERN IS ESTABLISHED BY T
YPING ADRS,RATE,MODE"CR".<15>]
D10:     .TXT     ]<12>THE EXISTING SENSOR SETTINGS MAY BE
EXAMINED BY TYPING ADRS"CR".<15>]
D12:     .TXT     ]<12>SENSOR ADDRESSES (ABBREVIATED ADR OR
 ADRS) MAY BE ENTERED:<15>]
D13:     .TXT     ]  1  SINGLY (ADR)<15>]
D14:     .TXT     ]  2  AS A RANGE (ADR1-ADR2)<15>]
D15:     .TXT     ]  3  AS A RANGE WITH AN INCREMENT (ADR1-
ADR2/INCR).<15>]
D16:     .TXT     ]WHENEVER "ADRS" IS CALLED FOR, ANY OF TH
ESE 3 FORMATS MAY BE USED.<15>]
D17:     .TXT     ]WHEN THE PROMPT INDICATES "ADR", A SINGL
E ADR MUST BE ENTERED.<15>]
D20      .TXT     ]<12>SAMPLING RATE IS A SINGLE DIGIT, 0 T
O 7.<15>]
D21:     .TXT     ]    0  DO NOT SAMPLE THIS SENSOR.<15>]
D22:     .TXT     ]    1  ONCE EVERY 10 SECONDS.<15>]
D23:     .TXT     ]    2  ONCE PER SECOND.<15>]
D24:     .TXT     ]    3  10/SEC.<15>]
```

```
D25:    .TXT    ]  4  40/SEC.<15>]
D26:    .TXT    ]  5  100/SEC.<15>]
D27:    .TXT    ]  6  250/SEC.<15>]
D28:    .TXT    ]  7  1000/SEC.<15>]
D29:    .TXT    ]DEFAULT IS RATE 2.<15>]
D30:    .TXT    ]<12>SAMPLING MODE IS A SINGLE LETTER, R
OR P.<15>]
D31:    .TXT    ]   R  REAL MODE (INSTANTANEOUS VALUE).<15>]
D32:    .TXT    ]   P  PEAK MODE (HIGHEST VALUE SINCE LAS
T READ).<15>]
D33:    .TXT    ]DEFAULT IS MODE P.<15>]
D40:    .TXT    ]SOME EXAMPLES:<15>]
D41:    .TXT    ]   100            TYPES THE PRESENT STATUS
 OF ADR 100<15>]
D42:    .TXT    ]   100,0          REMOVES ADR 100 FROM THE
 SAMPLING PATTERN (RATE 0)<15>]
D43:    .TXT    ]   100,5,R        SETS ADR 100 TO RATE 5,
REAL MODE<15>]
D44:    .TXT    ]   100-150,3,P  SETS ADR 100 THRU 150, INC
LUSIVE, TO RATE 3,<15>]
D45:    .TXT    ]                  PEAK MODE<15>]
D46:    .TXT    ]   100-150/3,0  REMOVES EVERY THIRD ADR
FROM 100 THRU 150 FROM<15>]
D47:    .TXT    ]                  THE SAMPLING PATTERN (10
0,103,106... 148)<15>]
D50:    .TXT    ]<12>THE MAXIMUM OVERALL SAMPLING RATE IS
 2850 SAMPLES PER SECOND.<15>]
D51:    .TXT    ]THIS MAY BE DIVIDED AMONG THE SENSORS IN
 ANY WAY BY THE<15>]
D52:    .TXT    ]OPERATOR.  THE LIMIT OF 2850 MAY NOT BE
EXCEEDED, BUT IS NOT<15>]
D53:    .TXT    ]NECESSARILY REACHED.  FOR EXAMPLE, ALL 2
55 SENSORS COULD BE<15>]
D54:    .TXT    ]SAMPLED AT RATE 3 (10/SEC) FOR A TOTAL O
F 2550 SAMPLES/SEC.<15>]
D55:    .TXT    ]OR THE PATTERN COULD INCLUDE 180 SENSORS
 AT RATE 3, ONE<15>]
D56:    .TXT    ]SENSOR AT RATE 7, 20 SENSORS AT RATE 1,
AND 54 SENSORS NOT<15>]
D57:    .TXT    ]SAMPLED (TOTAL 2802/SEC).  THREE SAMPLES
 AT RATE 7 (TOTAL<15>]
D58:    .TXT    ]3000 SAMPLES/SEC) IS NOT LEGAL.  THE PRO
GRAM KEEPS TRACK OF<15>]
D59:    .TXT    ]THE TOTAL SAMPLES/SEC REQUESTED AND PRIN
TS THE UPDATED VALUE<15>]
D60:    .TXT    ]AFTER EACH CHANGE.  IF A RATE IS REQUEST
ED WHICH WOULD EXCEED<15>]
D61:    .TXT    ]THE LIMIT, AN ERROR MESSAGE IS PRINTED A
ND THE SENSOR IN<15>]
D62:    .TXT    ]QUESTION REMAINS AT ITS OLD RATE.<12><15>]
;
;       ALARM LIMIT SELECTION
;
A51:    .TXT    ]<12>AN "ALARM CONDITION" EXISTS IF ANY S
ENSOR RETURNS A VALUE LESS<15>]
A52:    .TXT    ]THAN ITS ASSIGNED MINIMUM OR GREATER THA
N ITS ASSIGNED MAXIMUM.<15>]
A60:    .TXT    ]DEFAULT LIMITS ARE 0 (MIN) AND 1023 (MAX
) FOR ALL SENSORS. WITH<15>]
A61:    .TXT    ]THE DEFAULT LIMITS, NO ALARM CONDITION I
```

```
S POSSIBLE.<15>3
A64     .TXT    3<12>*INITIALIZE SETS ALL SENSORS TO DEFA
ULT LIMITS (0 MIN, 1023 MAX),<15>3
A65     .TXT    3 BUT DOES NOT AFFECT THE EXISTING RATE A
ND MODE.  ALARM LIMITS<15>3
A66:    .TXT    3 MAY BE EXAMINED OR MODIFIED.<15>3
A67     .TXT    3*MODIFY LEAVES THE EXISTING ALARM LIMITS
 INTACT.  ALARM LIMITS<15>3
A68     .TXT    3*CHECK LEAVES THE EXISTING ALARM LIMITS
INTACT.  SENSOR SETTINGS<15>3
A69:    .TXT    3 MAY BE EXAMINED, BUT NOT MODIFIED.<12><15>3
A54:    .TXT    3<12>ALARM LIMITS ARE ASSIGNED BY TYPING:
<15>3
A55     .TXT    3   ADRS,MIN,MAX"CR"  TO SET BOTH MIN AND
 MAX<15>3
A56:    .TXT    3   ADRS,MIN,"CR"     TO SET MIN AND LEAV
E MAX INTACT<15>3
A57:    .TXT    3   ADRS,,MAX"CR"     TO SET MAX AND LEAV
E MIN INTACT<15>3
A575:   .TXT    3ADRS IS AS DESCRIBED IN THE SAMPLING PAT
TERN SECTION.  THE MIN<15>3
A58:    .TXT    3MUST BE LESS THAN OR EQUAL TO TH
E MAX.  VALID SETTINGS ARE<15>3
A59:    .TXT    30 TO 1023, INCLUSIVE.<12><15>3
;
;       LOGGING
;
A70:    .TXT    3<12>LOGGING CONSISTS OF PRINTING, AT A S
PECIFIED TIME INTERVAL,<15>3
A71:    .TXT    3ONE VALUE FOR EACH SENSOR IN THE SAMPLIN
G PATTERN.  THE TIME<15>3
A72     .TXT    3INTERVAL MAY BE SPECIFIED IN HOURS OR MI
NUTES.  TYPE ONE OR <15>3
A74:    .TXT    3TWO DIGITS, FOLLOWED BY "H" FOR HOURS OR
 "M" FOR MINUTES.<12><15>3
;
;       MONITORING
;
A80     .TXT    3<12>MONITORING MAY REPLACE LOGGING.  ONE
 SPECIFIED SENSOR IS<15>3
A81:    .TXT    3PRINTED ABOUT ONCE EVERY TWO SECONDS.  E
NTER ZERO TO DISABLE<15>3
A82:    .TXT    3MONITORING OR AN ADR TO MONITOR THAT SEN
SOR.<12><15>3
;
;       ALARM "S"
;
A90:    .TXT    3<12>A SELECTED SENSOR MAY ALSO BE CONNEC
TED TO ALARM "S".  SO<15>3
A91:    .TXT    3THE ALARM WILL LATCH IF THE ASSIGNED LIM
ITS ARE EXCEEDED.<15>3
A92:    .TXT    3TYPE ZERO TO DISABLE THIS ALARM OR AN AD
R TO CONNECT THAT<15>3
A93:    .TXT    3SENSOR TO THE ALARM.<12><15>3
;
;   RECORDING MODE
;
B30:    .TXT    3<12>RECORDING OR STANDBY MODE MAY BE SEL
ECTED.  IF RECORDING IS<15>3
B31:    .TXT    3SELECTED, ALL VALUES SAMPLED ARE RECORDE
```

```
D ON TAPE. IF STANDBY IS<15>3
S72:    .TXT    3SELECTED, SAMPLING CONTINUES NORMALLY BU
T NO VALUES ARE RECORDED<15>3
S73:    .TXT    3UNLESS AN ALARM CONDITION EXISTS.  AN AL
ARM CONDITION TRIGGERS<15>3
S74     .TXT    3RECORDING FOR SOME NUMBER OF SECONDS. AF
TER WHICH THE PROGRAM<15>3
S75:    .TXT    3REVERTS TO STANDBY AGAIN.  IF STANDBY MO
DE IS SELECTED, THE<15>3
S76     .TXT    3DURATION (IN SECONDS) OF RECORDING MAY A
LSO BE SPECIFIED.<12><15>3
;
;       OPERATOR INTERVENTION - INSTRUCTIONS
;
S10:    .TXT    3<12>TYPE THE "ESC" KEY TO EXAMINE OR MODIFY
 OPTIONS.<15>3
;
;   OPERATOR INTERVENTION - INITIAL
;
S40:    .TXT    3THE SAMPLING PATTERN AND ALARM LIMITS
 MAY BE CHECKED AND ANY<15>3
S41:    .TXT    3OTHER SYSTEM PARAMETERS MAY BE MODIFI
ED WITHOUT HALTING DATA<15>3
S42     .TXT    3GATHERING.  HOWEVER, TO CHANGE THE SA
MPLING PATTERN OR ALARM<15>3
S43:    .TXT    3LIMITS, DATA GATHERING MUST CEASE AND
 ALL DATA WILL BE LOST<15>3
S44:    .TXT    3UNTIL CHANGES ARE COMPLETED.<12><15>3
;
;   OPERATOR INTERVENTION - CONTINUED
;
S50:    .TXT    3<12>WHEN THE "?" PROMPT APPEARS, INDICAT
E WHICH OPTION YOU WISH TO<15>3
S51:    .TXT    3MODIFY OR CHECK BY TYPING THE CODE LETTE
R LISTED BELOW, FOLLOWED<15>3
S52     .TXT    3BY "CR".  TYPE JUST "CR" WHEN YOU ARE DO
NE.<15>3
S60:    .TXT    3<12>  P  TO GET MINIMAL PROMPTING<15>3
S68:    .TXT    3  H  TO GET HELPFUL PRINTOUT<15>3
S61:    .TXT    3  D  DATE & TIME<15>3
S62:    .TXT    3  S  SAMPLING PATTERN<15>3
S63:    .TXT    3  A  ALARM LIMITS<15>3
S64     .TXT    3  L  LOGGING<15>3
S65:    .TXT    3  M  MONITORING & ALARM 3<15>3
S66:    .TXT    3  R  RECORDING MODE<15>3
S67:    .TXT    3  T  TAPE UNIT SELECTION<15>3
;
;    MAG TAPE
;
S70:    .TXT    3<12>MAG TAPE INITIAL SELECTION<12><15>3
S80:    .TXT    3<12>MAG TAPE MODIFICATION<12><15>3
;
;    INITIALIZATION OPTION
;
C10     .TXT    3<12>IF EVERYTHING IS OK, TYPE C TO BEGIN
 DATA GATHERING.  TO CHECK OR<15>3
C11     .TXT    3MODIFY ANYTHING, TYPE JUST "CR", AND
WHEN THE "?" PROMPT APPEARS<15>3
C12:    .TXT    3SELECT THE SECTION TO BE RE-RUN.  3
        .TITL   UB000   .BYTE POINTERS
```

```
          BYTE ADDRESSES

VE002
.GENERAL INSTRUCTIONS
          -27.
I000:     A00*2
          A01*2
          A015*2
          A017*2
          A03*2
          A04*2
          A02*2
          A025*2
          A05*2
          A06*2
          A07*2
          A08*2
          A09*2
          A11*2
          A12*2
          A13*2
          A14*2
          A15*2
          A16*2
          A17*2
          A18*2
          A19*2
          A20*2
.OPERATOR INTERVENTION - INSTRUCTIONS
          -1.
I010:     B10*2
.OPERATOR INTERVENTION - INITIAL
          -5.
I020:     B40*2
          B41*2
          B42*2
          B43*2
          B44*2
.INITIALIZATION OPTION
          -3.
I030:     C10*2
          C11*2
          C12*2
.OPERATOR INTERVENTION - CONTINUED
          -3.
I040      B50*2
          B51*2
          B52*2
.OPERATOR INTERVENTION - PROMPTING
          -9.
I050:     B60*2
          B68*2
          B61*2
          B62*2
          B63*2
          B64*2
          B65*2
          B66*2
          B67*2
.DATE & TIME
```

```
                -5.
I100            A30*2
                A31*2
                A32*2
                A33*2
                A34*2
/SAMPLING PATTERN
                -11.
I200,           E00*2
                E01*2
                D00*2
                D01*2
                D02*2
                D03*2
                D04*2
                D05*2
                D06*2
                D07*2
                A05*2
                -43.
I220,           D12*2
                D13*2
                D14*2
                D15*2
                D16*2
                D17*2
                D20*2
                D21*2
                D22*2
                D23*2
                D24*2
                D25*2
                D26*2
                D27*2
                D28*2
                D29*2
                D30*2
                D31*2
                D32*2
                D33*2
                D10*2
                D11*2
                D40*2
                D41*2
                D42*2
                D43*2
                D44*2
                D45*2
                D46*2
                D47*2
                D50*2
                D51*2
                D52*2
                D53*2
                D54*2
                D55*2
                D56*2
                D57*2
                D58*2
                D59*2
```

```
          C60*2
          C61*2
          C62*2
          -1.
Z34B      C10*2
.ALARM LIMITS
          -11.
Z30B:     A51*2
          A52*2
          A60*2
          A61*2
          A64*2
          A65*2
          A66*2
          A67*2
          A68*2
          A63*2
          A69*2
          -7.
Z32B      A54*2
          A55*2
          A56*2
          A57*2
          A575*2
          A58*2
          A59*2
.LOGGING
          -4.
Z40B:     A70*2
          A71*2
          A72*2
          A74*2
.MONITORING
          -3.
Z50B:     A80*2
          A81*2
          A82*2
.ALARM B
          -4.
Z60B:     A90*2
          A91*2
          A92*2
          A93*2
.RECORDING MODE
          -7.
Z70B:     B30*2
          B31*2
          B32*2
          B33*2
          B34*2
          B35*2
          B36*2
.TAPE ROUTINES
          -1.
Z80B:     B70*2
          -1.
Z82B      B80*2
          0
.INITIAL WELCOME
D00B      C00*2
```

```
C02B:   C02Y2
C03B:   C03Y2
        .TITL   VB000   ;CODE
;
;       TYPE GENERAL INSTRUCTIONS
;
VB00B:  Z00B    ;FIRST BYTE ADR
VB000:  STA     0,VB0A0
        STA     1,VB0A1         ;SAVE REGISTERS
        STA     2,VB0A2
        STA     3,VB0A3
;
        LDA     0,C00B          ;TYPE WELCOME
        JSR     @P.CR
;
        LDA     0,C02B          ;TYPE QUESTION
        JSR     @P.CR
;
        LDA     0,C03B          ;TYPE ALTERNATIVE
        JSR     @P.CR
        JSR     @R.LN           ;READ RESPONSE
        JMP     VB005           ;JUST "CR" - MINIMAL PROMPTING
;
        LDA     0,IBUF
        JSR     @GT.CH          ;GET FIRST CHAR
;
        LDA     0,QRH           ;CHECK FOR "H"
        SUB     0,1,SZR
VB005:  SUB     0,0             ;MINIMAL PROMPTING
        STA     0,VBFLG
;
        LDA     2,VB00B         ;GET ADR BYTE POINTERS
        JMP     VB905           ;GO PRINT LINES
;
;       TYPE OPERATOR INTERVENTION - INSTRUCTIONS
;
VB01B:  Z01B    ;FIRST BYTE ADR
VB010:  STA     2,VB0A2
        LDA     2,VB01B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
;       TYPE OPERATOR INTERVENTION - INITIAL
;
VB02B:  Z02B    ;FIRST BYTE ADR
VB020:  STA     2,VB0A2
        LDA     2,VB02B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
;       TYPE INITIALIZATION OPTION
;
VB03B:  Z03B    ;FIRST BYTE ADR
VB030:  STA     2,VB0A2
        LDA     2,VB03B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
;       TYPE OPERATOR INTERVENTION - CONTINUED
;
VB04B:  Z04B    ;FIRST BYTE ADR
VB040:  STA     2,VB0A2
        LDA     2,VB04B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
        TYPE OPERATOR INTERVENTION - PROMPTING
;
VB06B   Z06B    ;FIRST BYTE ADR
VB060   STA     2,VB0A2
        LDA     2,VB06B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
        TYPE DATE & TIME
;
VB10B   Z10B    ;FIRST BYTE ADR
VB100   STA     2,VB0A2
        LDA     2,VB10B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
        TYPE SAMPLING PATTERN SELECTION
;
VB20B   Z20B    ;FIRST BYTE ADR
VB200   STA     2,VB0A2
        LDA     2,VB20B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
VB22B   Z22B    ;FIRST BYTE ADR
VB220   STA     2,VB0A2
        LDA     2,VB22B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
VB24B   Z24B    ;FIRST BYTE ADR
VB240   STA     2,VB0A2
        LDA     2,VB24B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
        TYPE ALARM LIMIT SELECTION
;
VB30B   Z30B    ;FIRST BYTE ADR
VB300   STA     2,VB0A2
        LDA     2,VB30B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
VB32B   Z32B    ;FIRST BYTE ADR
VB320   STA     2,VB0A2
        LDA     2,VB32B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
        TYPE LOGGING
;
VB40B   Z40B    ;FIRST BYTE ADR
VB400   STA     2,VB0A2
        LDA     2,VB40B         ;GET AUR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
        TYPE MONITORING
;
VB50B   Z50B    ;FIRST BYTE ADR
VB500   STA     2,VB0A2
        LDA     2,VB50B         ;GET ADR BYTE POINTERS
        JMP     VB900           ;GO PRINT LINES
;
        TYPE ALARM "B"
;
VB60B   Z60B    ;FIRST BYTE ADR
VB600   STA     2,VB0A2
```

```
        LDA     2,UB60B         ;GET ADR BYTE POINTERS
        JMP     UB900           ;GO PRINT LINES
;
;       RECORDING MODE
;
UB70B   I70B    ;FIRST BYTE ADR
UB700   STA     2,UB0A2
        LDA     2,UB70B         ;GET ADR BYTE POINTERS
        JMP     UB900           ;GO PRINT LINES
;
;       TYPE TAPE ROUTINE - INITIAL
;
UB80B   I80B    ;FIRST BYTE ADR
UB800   STA     2,UB0A2
        LDA     2,UB80B         ;GET ADR BYTE POINTERS
        JMP     UB900           ;GO PRINT LINES
;
;       TYPE TAPE ROUTINE - MODIFY
;
UB82B   I82B    ;FIRST BYTE ADR
UB820   STA     2,UB0A2
        LDA     2,UB82B         ;GET ADR BYTE POINTERS
        JMP     UB900           ;GO PRINT LINES
        .TITL   UB900   ;COMMON PROCESSING
;
;       TYPE THE LINES
;
UB900   STA     0,UB0A0
        STA     1,UB0A1
        STA     3,UB0A3         ;SAVE REGISTERS
;
UB905   LDA     0,UBFLG         ;CHECK WHETHER FULL PRINTOUT DESIRED
        MOV     0,0,SNR
        JMP     UB990
;LOOP TO TYPE LINES
        LDA     1,-1,2          ;GET LINE COUNT
UB910   LDA     0,0,2           ;GET BYTE POINTER
        MOV     0,0,SNR
        JMP     UB980           ;END ON ZERO BYTE POINTER
        JSR     @P.CR           ;TYPE A LINE
        INC     2,2             ;INCR BYTE POINTER ADR
        INC     1,1,SZR         ;COUNT LOOP
        JMP     UB910
;RETURN
UB990   LDA     0,UB0A0
        LDA     1,UB0A1
        LDA     2,UB0A2
        JMP     @UB0A3
;STORAGE
UB0A0   .-.
UB0A1   .-.
UB0A2:  .-.
UB0A3   .-.
DDW     110
UBFLG   .-.
        .END
;
```

# APPENDIX B

# PLAYBACK PROGRAM

# LISTINGS

```
TYPE PB.FR
C       PROGRAM PB
        DOUBLE PRECISION CMEAN,RMS
        CHANTASK 20,5
        EXTERNAL PBESC,PBMN,PBLG,PBDP
        COMMON /CIARY/ IAR(3)
        COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1          ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
        COMMON /CHANL/ ICTTO,ICTTI,ICMTO,ICMT1,JREAD,ISTS,IER,NWORD
        COMMON /CSTAT/ NREC,CNUM(256),CMIN(256),CMAX(256),CMEAN(256),
     1          MIN(256),MAX(256),RMS(256)
        COMMON /CTREC/ JDATE(6),IMS,ITYP,IN(4088),NIN
        COMMON /CTPST/ LADR1,LADR2,LINCR,LTFOR,LPFOR,LALL
        COMMON /CFLAG/ IFMON,IFPRO,IFEOF,IFEOT,IFTYP
        COMMON /CDATE/ KDATE(6),KDAT1(6),KDAT2(6)
        COMMON /CMAIN/ IFL(4),IUN(2)
        DATA IAR /-1,3,40030K/
        DATA ICTTO,ICTTI/3,2/
        DATA IFL(1)/"MTO:0"/
        DATA IUN(1)/"MTO"/
        DATA JREAD/0/,IHIAD/256/
C
        CALL RESET
        CALL OPEN (ICTTO,"$TTO",IAR,IER)
        IF (IER.NE.1) TYPE "<6>TTO ERR",IER
        CALL OPEN (ICTTI,"$TTI",1,IER)
        IF (IER.NE.1) TYPE "<6>TTI ERR",IER
        TYPE " * * * MV ARCTIC PLAYBACK PROGRAM * * *"
C       --I KNOW IT LOOKS CRAZY, BUT THIS IS NECESSARY
C       --TO PRINT LINES > 80 CHARACTERS!!
        CALL CLOSE (ICTTO,IER)
        CALL OPEN (ICTTO,"$TTO",IAR,IER)
100     WRITE(ICTTO,10)
   10   FORMAT(" <6>MOUNT TAPE FOR READING,"Z)
        JOPT=1
        IFEOF=0
        IFEOT=0
        IFPRO=0
        WRITE (ICTTO,11)
   11   FORMAT (" <6> TYPE UNIT # (0 OR 1)"Z)
        CALL PBACC (ICMTO,0,1)
        IF (ICMTO.EQ.1) GO TO 110
        IUN(2)="0"
        IFL(2)="0:"
        GO TO 120
110     IUN(2)="1"
        IFL(2)="1:"
C  INITIALIZE TAPE
120     CALL INIT(IUN,0,IER)
        IF (IER.NE.1) GO TO 130
C  GET & OPEN FILE
        WRITE(ICTTO,14)
   14   FORMAT(" <6>TYPE FILE ## OR 00: "Z)
        READ (ICTTI,15) IFL(3)
   15   FORMAT (A2)
        IFL(4)=0
        CALL MTOPD(ICMTO,IFL,0,IER)
        IF (IER.EQ.1) GO TO 160
C  ERROR - CLOSE & RELEASE & TRY AGAIN
130     TYPE "<6>FORT ERR",IER
140     CALL CLOSE(ICMTO,IER)
        CALL RLSE(IUN,IER)
        GO TO 100
C  READ HEADER
150     IF (IFEOF.EQ.0) CALL MTDIO(ICMTO,30000K,JDATE,ISTS,IER,NWORD)
160     CALL PBMTH($290)
C  USER OPTIONS
170     TYPE '<6>1 TO START HERE'
        TYPE '<6>2 TO SEARCH FOR START TIME'
        TYPE '<6>3 TO SEARCH FOR ALARMS'
        TYPE '<6>4 TO MODIFY ALARMS'
        CALL PBACC (JOPT,1,4)
        KOPT = JOPT-1
        GO TO (300,180,200,175),JOPT
C  MODIFY ALARMS
175     CALL PBALM
        GO TO 170
C  GET START TIME
180     WRITE (ICTTO,18)
   18   FORMAT (' <6>YR,MO,DA: 'Z)
182     READ (ICTTI,ERR=184,END=184) KDAT1(1),KDAT1(2),KDAT1(3)
        GO TO 190
184     WRITE (ICTTO,17)
   17   FORMAT (' <6>?: 'Z)
        GO TO 182
C
190     WRITE (ICTTO,19)
   19   FORMAT (' <6>HR,MI,SC: 'Z)
192     READ (ICTTI,ERR=194,END=194) KDAT1(4),KDAT1(5),KDAT1(6)
        GO TO 200
194     WRITE (ICTTO,17)
        GO TO 192
C
C  SEARCH FOR START POINT
C
200     CALL MTDIO(ICMTO,JREAD,JDATE,ISTS,IER,NWORD)
        CALL PBMTS($270)
        NIN=NWORD-8
        GO TO (210,230),KOPT
C
C   SEARCH FOR START TIME
210     DO 220 I1=1,6
        IF (JDATE(I1).LT.KDAT1(I1)) GO TO 200
220     CONTINUE
C  FOUND START TIME
        WRITE (ICTTO,22) JDATE
   22   FORMAT (' <6>RECORD FOUND:'I3'Y'I3'M'I3'D'I3'H'I3'M'I3'S')
        IFPRO=0
        GO TO 300
C
C  SEARCH FOR ALARM LIMITS
230     DO 240 I1=1,NIN,2
        IADR=IAND(IN(I1),777K)
        IVAL=IAND(1777K,IN(I1+1))
        IF (IVAL.LT.ITMIN(IADR)) GO TO 250
        IF (IVAL.GT.ITMAX(IADR)) GO TO 250
240     CONTINUE
        GO TO 200
C  ALARM EXCEEDED
250     WRITE (ICTTO,25) IADR
   25   FORMAT (' <6>'I3' LIMIT EXCEEDED')
        IFPRO=0
        GO TO 300
```

```
C  EOF ENCOUNTERED
270     TYPE '<6>EOF'
        IF (KOPT.EQ.1) TYPE '<6>START TIME NOT FOUND'
        IF (KOPT.EQ.2) TYPE '<6>NO ALARMS EXCEEDED'
        TYPE '<6>1 TO CONTINUE SEARCHING NEXT FILE'
        TYPE '<6>2 FOR NEW SEARCH'
        TYPE '<6>3 FOR NEW TAPE'
        CALL PBACC (IOPT,1,3)
        GO TO (280,160,140),IOPT
C  CONTINUE SEARCHING
280     CALL PBMTH($290)
        CALL PBALM
        GO TO 200
C  EOT
290     GO TO (100,292,293),JOPT
292     TYPE '<6>START TIME NOT FOUND'
        GO TO 100
293     TYPE '<6>NO ALARMS EXCEEDED'
        GO TO 100
C
C  PROCESS TAPE
C
300     CALL PB2DT
        CALL PBSTI
        TYPE '<6>1 MONITORING'
        TYPE '<6>2 DUMP'
        CALL PBACC (IOPT,1,2)
C
        CALL ITASK(PBESC,2,2,IER)
        IF (IER.EQ.1) GO TO 320
        TYPE '<6>ESC IER=',IER
        GO TO 380
C
320     GO TO (330,350),IOPT
330     CALL ITASK(PBMN,3,3,IER)
        IFMON=1
        GO TO 360
350     CALL ITASK(PBDP,3,3,IER)
        IFMON=0
C
360     IF (IER.EQ.1) GO TO 370
        TYPE '<6>OPT',IOPT,' ERR=',IER
        GO TO 380
370     CALL SUSP
380     CALL AKILL(2)
        CALL AKILL(3)
        IF (IFEOT.EQ.1) GO TO 100
        TYPE '<6>1 NEW OUTPUT'
        TYPE '<6>2 NEW SEARCH'
        TYPE '<6>3 NEW FILE'
        TYPE '<6>4 NEW TAPE'
        CALL PBACC (IOPT,1,4)
        GO TO (300,170,150,140),IOPT
        END
R
```

```
TYPE PB2DT.FR
        SUBROUTINE PB2DT
        COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
        COMMON /CDATE/ KDATE(6),KDAT1(6),KDT2A(3),KDT2B(3)
        WRITE(ICTTO,11)
   11   FORMAT(" <6>END DATE (Y OR N)? "Z)
        IF (IPBYN(I1)) 110,110,120
110     KDT2A(1)=2000
        RETURN
C
120     WRITE (ICTTO,12)
   12   FORMAT (" <6>YR,MO,DA: "Z)
122     READ (ICTTI,END=124,ERR=124) KDT2A
        GO TO 130
124     WRITE (ICTTO,15)
   15   FORMAT (" <6>?: "Z)
        GO TO 122
130     WRITE (ICTTO,13)
   13   FORMAT (" <6>HR,MI,SC: "Z)
132     READ (ICTTI,END=134,ERR=134) KDT2B
        GO TO 140
134     WRITE (ICTTO,15)
        GO TO 132
140     RETURN
        END
R
```

```
      SUBROUTINE PBACC (IN,MIN,MAX)
C  DO ACCEPT FUNCTION WITH ERROR CHECKING
      COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
      WRITE (ICTTO,11)
  11  FORMAT (" <6>: "Z)
110   READ (ICTTI,ERR=120,END=130) IN
      IF (IN.LT.MIN .OR. IN.GT.MAX) GO TO 120
      RETURN
120   WRITE (ICTTO,12)
  12  FORMAT (" <6>?: "Z)
      GO TO 110
130   TYPE "END="
      RETURN
      END
R
```

```
TYPE PBDP.FR
        TASK PBDP        ;PLAYBACK DUMP
        COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1         ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
        COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
        COMMON /CTREC/ JDATE(6),IMS,ITYP,IN(4088),NIN
        COMMON /CFLAG/IFMON,IFPRO,IFEOF,IFEOT,IFTYP
        COMMON /CDATE/ KDATE(6),KDAT1(6),KDAT2(6)
C   TYPING
110     WRITE(ICTTO,11)
   11   FORMAT(" <6>TYPE? "Z)
        IFTYP=IPBYN(I1)
C   PROCESS
120     CALL PBMTR($110)
        IF (IFTYP.EQ.0) GO TO 120
        WRITE (ICTTO,12) KDATE
   12   FORMAT (" <6>"I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S")
        DO 130 I1=1,NIN,2
        IADR=IAND(IN(I1),777K)
        IVAL=IAND(IN(I1+1),1777K)
        IFLAG="  "
        IF (IVAL.LT.ITMIN(IADR)) IFLAG="* "
        IF (IVAL.GT.ITMAX(IADR)) IFLAG="* "
        IF (IVAL.NE.IN(I1+1)) IFLAG="**"
        VAL=PBVAL(IVAL,IADR,UNIT)
        WRITE (ICTTO,13) IADR,VAL,IFLAG,UNIT
   13   FORMAT (" <6>"I3,F9.2,A2,1X,S4)
130     CONTINUE
        GO TO 120
        END
```

```
TYPE PBESC.FR
        TASK PBESC
        COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
        COMMON /CFLAG/IFMON,IFPRO,IFEOF,IFEOT,IFTYP
        COMMON /CDATE/KDATE(6),KDAT1(6),KDAT2(6)
        I1=1
A       STA     0,PBEA0
A       STA     1,PBEA1
A       STA     2,PBEA2
A       STA     3,PBEA3
A       LDA     0,QRESC
A        .SYSTEM
A        .WCHAR
A       LDA     2,PBEA2
A       LDA     0,PBEA0
A       LDA     3,PBEA3
A       MOVZL#  1,1,SNC
A       JMP     PBE10
A;WAIT TERMINATED
A       LDA     1,PBEA1
        CALL KILL
AQRESC: 33
APBEA0: .-.
APBEA1: .-.
APBEA2: .-.
APBEA3: .-.
A;ESC RECEIVED
APBE10: LDA      1,PBEA1
        CALL ABORT(3,IER)
        IF (IER.NE.1) TYPE "ABORT ERR",IER
        TYPE "<6>ESC"
        WRITE(ICTTO,20) (KDATE(L),L=1,6)
20      FORMAT(" <6>TAPE PROCESSED TILL:"I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S")
        IF (IFMON.EQ.1) CALL PBMNF
        CALL PBSTT
        CALL ARDY(0)
        CALL KILL
        END
R
```

```
TYPE PBKES.FR
        SUBROUTINE PBKES
        I1=1
        STA     0,PBKA0
A       STA     1,PBKA1
A       STA     2,PBKA2
A       STA     3,PBKA3
A       LDA     0,M1
A       .SYSTEM
A       .WCHAR
A       LDA     2,PBKA2
A       LDA     1,PBKA1
A       LDA     0,PBKA0
A       LDA     3,PBKA3
A       JMP     PBK10
APBKA0: .-.
APBKA1: .-.
APBKA2: .-.
APBKA3: .-.
AM1:    -1.
APBK10: JMP     .+1
C
        RETURN
        END
R
```

```
TYPE PBMTH.FR
      SUBROUTINE PBMTH (IRTN)
      COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1          ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
      COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
      COMMON /CFLAG/ IFMON,IFPRO,IFEOF,IFEOT,IFTYP
      COMMON /CDATE/ KDATE(6),KDAT1(6),KDAT2(6)
      COMMON /CMAIN/ IFL(4),IUN(2)
      COMMON /CMTH/ CONST(0:7)
      DATA CONST/0.,.1,1.,10.,40.,100.,250.,1000./
C  READ HEADER
      CALL MTDIO(ICMT0,JREAD,IDATE,ISTS,IER,NWORD)
      CALL PBMTS ($150)
      IF (NWORD.NE.1546) GO TO 140
C  REPORT DATE, SAMPLES/SEC
      SUM=0.0
      DO 110 I1=1,IHIAD
      I2=ISHFT(IAND(16000K,ITAB1(I1)),-10)
110   SUM=SUM+CONST(I2)
      WRITE (ICTTO,11) IDATE,SUM
   11 FORMAT (/" <6>FILE START"I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S"/
     1 " <6>"F6.1" SAMPLES/SEC")
      IF (IHIAD.LE.256) GO TO 120
      TYPE "<6>CAN'T HANDLE ADR 257-511"
      IHIAD=256
C  INITIALIZE
120   IFEOF=0
      IFEOT=0
      IFPRO=1
      DO 130 I1=1,6
130   KDATE(I1)=IDATE(I1)
      RETURN
C
C   NO HEADER
C
140   TYPE "<6>NO HEADER. NWORD=",NWORD
C
C  EOT
C
150   TYPE "<6>EOT"
      IFEOT=1
      IFEOF=1
      CALL CLOSE(ICMT0,IER)
      CALL RLSE(IUN,IER)
      RETURN IRTN
      END
R
```

```
TYPE PBMTR.FR
      SUBROUTINE PBMTR (IRTN)
      COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
      COMMON /CTREC/ JDATE(6),IMS,ITYP,IN(4088),NIN
      COMMON /CFLAG/IFMON,IFPRO,IFEOF,IFEOT,IFTYP
      COMMON /CDATE/ KDATE(6),KDAT1(6),KDAT2(6)
C
C  NORMAL READ AND CHECK
C
      IF (IFEOF.EQ.1) CALL PBMTH($190)
      IF (IFPRO.EQ.0) GO TO 120
      IFPRO=0
      CALL MTDIO(ICMT0,JREAD,JDATE,ISTS,IER,NWORD)
      CALL PBMTS($180)
      NIN=NWORD-8
120   DO 130 I1=1,6
      IF (JDATE(I1)-KDAT2(I1)) 140,130,160
130   CONTINUE
C  TIME OK
140   DO 145 I1=1,6
145   KDATE(I1)=JDATE(I1)
      CALL PBSTG
      IFPRO=1
      RETURN
C
C  TIME EXCEEDS END TIME
C
160   WRITE (ICTTO,16) KDATE,KDAT2,JDATE
      IF (IFMON.EQ.1) CALL PBMNF
      CALL PBSTT
      WRITE(ICTTO,15)
  15  FORMAT(" <6>CONTINUE? "Z)
      IF (IPBYN(I1)) 190,190,170
C  CONTINUE PRESENT TASK - GET NEW END TIME
170   CALL PB2DT
      RETURN IRTN
C
C  EOF
C
180   TYPE "<6>EOF"
      WRITE (ICTTO,18) KDATE
      IFEOF=1
      IF (IFMON.EQ.1) CALL PBMNF
      CALL PBSTT
C
C  FINAL PROCESSING
C
190   CALL PBKES
      CALL ARDY(0)
      CALL KILL
C
  16  FORMAT (" <6>LAST TIME"I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S"/
     1 " <6>END TIME "I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S"/
     2 " <6>THIS TIME"I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S")
  18  FORMAT(" <6>LAST TIME"I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S")
      END
R
```

```
TYPE PBMTS.FR
      SUBROUTINE PBMTS (IRTN)
C
      COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
C   TEST BITS THAT SHOULD BE ON
      IF(.NOT.ITEST(ISTS,11)) TYPE " DENSITY"
      IF(.NOT.ITEST(ISTS,6)) TYPE " 9-TRACK"
      IF(.NOT.ITEST(ISTS,0)) TYPE " READY"
C   TEST BITS THAT SHOULD BE OFF
      IF(ITEST(ISTS,14)) TYPE " DATA LATE"
      IF(ITEST(ISTS,13)) TYPE " REWINDING"
      IF(ITEST(ISTS,12)) TYPE " ILLEGAL"
      IF(ITEST(ISTS,10)) TYPE " PARITY"
      IF(ITEST(ISTS,9)) TYPE " EOT"
      IF(ITEST(ISTS,7)) TYPE " LOAD PT"
      IF(ITEST(ISTS,5)) TYPE " BAD TAPE"
      IF(ITEST(ISTS,4)) TYPE " BIT 11"
      IF(ITEST(ISTS,3)) TYPE " BIT 12"
      IF(.NOT.ITEST(ISTS,1)) TYPE " ODD CHAR"
C   CHECK EOF
      IF (.NOT.ITEST(ISTS,8))GO TO 110
C
      IF (IER.NE.0) TYPE " FORT ERR",IER
      IF (NWORD.NE.1) TYPE " NWORD=",NWORD
      RETURN IRTN
C
110   IF (IER.NE.1 .AND. IER.NE.0) TYPE " FORT ERR",IER
      IF (NWORD.LE.6) TYPE " NWORD=",NWORD
      RETURN
      END
R
```

```
TYPE PBALM.FR
      SUBROUTINE PBALM
C
C  EXAMINE OR MODIFY ALARM LIMITS
C
      COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1         ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
      COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
C
      TYPE "<6>EXAMINE OR MODIFY ALARM LIMITS.  TYPE:"
      TYPE "<6>0 TO EXIT"
      TYPE "<6>1 TO EXAMINE ONE ADR"
      TYPE "<6>2 TO EXAMINE A RANGE"
      TYPE "<6>3 TO MODIFY ONE ADR"
      TYPE "<6>4 TO MODIFY A RANGE"
140   WRITE (ICTTO,40)
   40 FORMAT (" <6>(0-4): "Z)
      READ (ICTTI,ERR=145,END=200) N
      IF (N.EQ.0) GO TO 200
      N1=0
      IF (N.LT.5) GO TO (151,152,153,154),N
145   TYPE "<6>ERROR"
      GO TO 140
C  GET SINGLE ADR
151   WRITE (ICTTO,51)
   51 FORMAT (" <6>ADR: "Z)
      READ(ICTTI,ERR=145,END=200) LADR1
      LADR2=LADR1
      GO TO 170
C  GET ADR RANGE
152   WRITE (ICTTO,52)
   52 FORMAT (" <6>A1,A2: "Z)
      READ (ICTTI,ERR=145,END=200) LADR1,LADR2
      GO TO 160
C  GET ADR,MIN,MAX
153   WRITE (ICTTO,53)
   53 FORMAT (" <6>ADR,MN,MX: "Z)
      READ (ICTTI,ERR=145,END=200) LADR1,LMIN,LMAX
      LADR2=LADR1
      GO TO 155
C  GET ADR RANGE,MIN,MAX
154   WRITE (ICTTO,54)
   54 FORMAT (" <6>A1,A2,MN,MX: "Z)
      READ (ICTTI,ERR=145,END=200) LADR1,LADR2,LMIN,LMAX
155   N1=1
      IF(LMAX.LT.LMIN) GO TO 145
C
160   IF (LADR1.GT.LADR2) GO TO 145
C
C  LOOP TO PROCESS
C
170   DO 190 I1=LADR1,LADR2
      IRAT=ISHFT(IAND(16000K,ITAB1(I1)),-10)
      IF (IRAT.EQ.0) GO TO 190
      IF (N1.EQ.0) GO TO 180
      ITMIN(I1)=LMIN
      ITMAX(I1)=LMAX
      IF (I1.GT.LADR1 .AND. I1.LT.LADR2) GO TO 190
C  PRINT RESULTS
180   IM="P "
```

```
      IF (IAND(1000K,ITAB1(I1)).NE.0) IM="R "
      XMIN=PBVAL(ITMIN(I1),I1,UNIT)
      XMAX=PBVAL(ITMAX(I1),I1,UNIT)
      WRITE (ICTTO,26) I1,IRAT,IM,ITMIN(I1),ITMAX(I1),XMIN,XMAX,UNIT
26    FORMAT ("<6>"I3,I2,1X,A2,2I6,2F9.2,2X,S4)
C
190   CONTINUE
      GO TO 140
200   RETURN
      END
R
```

```
TYPE  PBMN.FR
      TASK PBMN       ;PLAYBACK MONITORING
      DIMENSION MONIT(10),JSTS(10),VALUE(10)
      DOUBLE PRECISION DRATE,UNIT,PUNIT(10)
      COMMON / CMNF/ YCNT,YCMIN,YMIN,YCMAX,YMAX,YMEAN,YRMS,
     1          IMINY,IMAXY,IADR
      COMMON /CPBMN/ DRATE(7),ASTR(0:1)
      COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1          ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
      COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
      COMMON /CTREC/ JDATE(6),IMS,ITYP,IN(4088),NIN
      COMMON /CFLAG/IFMON,IFPRO,IFEOF,IFEOT,IFTYP
      COMMON /CDATE/ KDATE(6),KDAT1(6),KDAT2(6)
      DATA DRATE(1)/"1/10 SEC"/
      DATA DRATE(2)/"1/SEC   "/
      DATA DRATE(3)/"10/SEC  "/
      DATA DRATE(4)/"40/SEC  "/
      DATA DRATE(5)/"100/SEC "/
      DATA DRATE(6)/"250/SEC "/
      DATA DRATE(7)/"1000/SEC"/
      DATA ASTR(0)/"PEAK"/,ASTR(1)/"REAL"/
C  INITIALIZE MONITORING STATISTICS
      YCNT=0.
      YCMIN=0.
      YCMAX=0.
      YMEAN=0.
      YRMS=0.
      IMINY=1023
      IMAXY=0
      NMON=0
      NREQ=0
      MINADR=0
      MINRAT=7
C  GET ADDRESS/ES
      WRITE(ICTTO,401)
401   FORMAT(" <6>HOW MANY CHANNELS?"Z)
      CALL PBACC(NMON,1,10)
      TYPE "<6>ENTER ADDRESSES ONE AT A TIME"
      DO 402 KK=1,NMON
110   WRITE (ICTTO,11)
  11  FORMAT (" <6>ADR"Z)
      CALL PBACC (IADR,1,IHIAD)
C  CHECK RATE
      IRAT=ISHFT(IAND(16000K,ITAB1(IADR)),-10)
      IF (IRAT.NE.0) GO TO 130
      TYPE "<6>ADR NOT SAMPLED"
      GO TO 110
C  REPORT STATUS
C
130   MONIT(KK)=IADR
      IF (MINRAT.LT.IRAT) GO TO 131
      MINRAT=IRAT
      MINADR=IADR
131   MODE=ISHFT(IAND(1000K,ITAB1(IADR)),-9)
      WRITE (ICTTO,13) IADR,DRATE(IRAT),ASTR(MODE)
  13  FORMAT (" <6>#"I3" SAMPLED AT ",S8," IN ",S4," MODE.")
      XMIN=PBVAL(ITMIN(IADR),IADR,UNIT)
      XMAX=PBVAL(ITMAX(IADR),IADR,UNIT)
      PUNIT(KK)=UNIT
402   CONTINUE
```

```
      IF (NMON.GT.1) GOTO 151
      WRITE (ICTTO,14) ITMIN(IADR),ITMAX(IADR),XMIN,XMAX,UNIT
  14  FORMAT (" <6>RECORDED LIMITS"2I6,2F9.2,1X,S4)
C  GET MONITORING LIMITS
      WRITE (ICTTO,15)
  15  FORMAT ( " <6>MONITORING LIMITS: "Z)
140   READ (ICTTI,END=144,ERR=142) IYMIN,IYMAX
      YMIN=PBVAL(IYMIN,IADR,UNIT)
      YMAX=PBVAL(IYMAX,IADR,UNIT)
      WRITE (ICTTO,16) YMIN,YMAX,UNIT
  16  FORMAT (" <5>"2F9.2,1X,S4)
      GO TO 150
142   WRITE (ICTTO,17)
  17  FORMAT (" <6>?: "Z)
      GO TO 140
144   TYPE "<6>"
      YMIN=XMIN
      YMAX=XMAX
      GO TO 150
C  GET SCAN RATE
      IF(NMON.EQ.1) GOTO 150
151   WRITE(ICTTO,410) DRATE(MINRAT),
410   FORMAT(" <6>LOWEST RATE="S8)
      TYPE "<6>BASED ON THIS RATE,"
150   WRITE (ICTTO,18)
  18  FORMAT (" <6>MONITOR EVERY NTH READING.  N"Z)
      CALL PBACC (N,1,30000)
      N1=1
C
      TYPE " <15>"
      WRITE(ICTTO,300) (KDATE(L),L=1,3)
[illegible]   FORMAT(" <6>"I2"Y"I2"M"I2"D"30X"CHANNEL NUMBER")
      WRITE(ICTTO,301) (MONIT(L),L=1,NMON)
301   FORMAT(" <6> TIME ",10I12)
      WRITE(ICTTO,302) (PUNIT(M),M=1,NMON)
302   FORMAT(" <6> H: M: S ",10(4X,"[",S6,"]"))
      TYPE " <6><15>"
C
C  LOOP TO MONITOR
C
C     SINGLE ADDRESS
      IF (NMON.GT.1) GO TO 291
200   CALL PBMTR($160)
      N2=0
C     PROCESS BUFFER
C
C  TYPE OR NO?
160   WRITE(ICTTO,19)
  19  FORMAT(" <6>TYPE? "Z)
      IFTYP=IPBYN(I1)
C
      DO 290 I1=1,NIN,2
      IF (IAND(IN(I1),777K) .NE. IADR) GO TO 290
      N1=N1-1
      IF (N1.GT.0) GO TO 290
C  MONITOR THIS READING
      N1=N
      IVAL=IAND(IN(I1+1),1777K)
      IFLAG="  "
      YCNT=YCNT+1.
```

```
        VAL=PBVAL(IVAL,IADR,UNIT)
        IF (VAL.GE.YMIN) GO TO 240
        YCMIN=YCMIN+1.
        IFLAG="* "
240     IF (VAL.LE.YMAX) GO TO 250
        YCMAX=YCMAX+1.
        IFLAG="* "
250     IF (IVAL.LT.IMINY) IMINY=IVAL
        IF (IVAL.GT.IMAXY) IMAXY=IVAL
        IF (IVAL.EQ.IN(I1+1)) GO TO 260
        IFLAG="**"
260     YRMS=YRMS+VAL*VAL
        YMEAN=YMEAN+VAL
        IF (IFTYP.EQ.0) GO TO 290
        IF (N2.EQ.0) WRITE (ICTTO,26) VAL,IFLAG,UNIT,JDATE
        IF (N2.NE.0) WRITE (ICTTO,27) VAL,IFLAG
        N2=1
C
   26   FORMAT (" <6>"F9.2,A2,1X,S4,I3"Y"I3"M"I3"D"I3"H"I3"M"I3"S")
   27   FORMAT (" <6>"F9.2,A2)
290     CONTINUE
        GO TO 200
C
C        MULTI-ADDRESSES
291     CALL PBMTR($160)
        N2=0
C  PROCESS BUFFER
        DO 292 I1=1,NIN,2
        IADR=IAND(IN(I1),777K)
        IF (IADR.EQ.MINADR)N1=N1-1
        IF (N1.GT.0.AND.NREQ.EQ.0) GO TO 292
        DO 201 J2=1,NMON
        NOW=J2
        IF (IADR.EQ.MONIT(J2)) GO TO 202
201     CONTINUE
        GO TO 292
202     IF (NREQ.NE.0) GOTO 500
C  START GATHERING INFO FOR P/O I.E. N1=0
        DO 203 K1=1,NMON
203     JSTS(K1)=0
        JSTS(NOW)=1
        NREQ=NMON-1
        IVAL=IAND(IN(I1+1),1777K)
        VALUE(NOW)=PBVAL(IVAL,IADR,UNIT)
        N1=N
        GO TO 292
C  VALUE REQUIRED FOR PRINTING FOUND
500     IF(JSTS(NOW).EQ.1) GO TO 292
        IVAL=IAND(IN(I1+1),1777K)
        VALUE(NOW)=PBVAL(IVAL,IADR,UNIT)
        JSTS(NOW)=1
        NREQ=NREQ-1
        IF (NREQ.NE.0) GOTO 292
C  TYPE THIS SET OF DATA
        WRITE(ICTTO,501) (KDATE(L),L=4,6),(VALUE(K),K=1,NMON)
501     FORMAT(" <6>"I2":"I2":"I2,10F12.2)
292     CONTINUE

        GO TO 291
        END
```

```
R
TYPE PBMNF.FR
        SUBROUTINE PBMNF           ;TYPE MONITORING STATISTICS
        DOUBLE PRECISION UNIT
        COMMON /CMNF / YCNT,Y(4),YMEAN,YRMS,IMINY,IMAXY,IADR
        COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD

        XMIN=PBVAL(IMINY,IADR,UNIT)
        XMAX=PBVAL(IMAXY,IADR,UNIT)
        XMEAN=YMEAN/YCNT
        XRMS=SQRT(YRMS/YCNT)
        WRITE (ICTTO,10) YCNT,UNIT,Y,XMIN,XMAX,XMEAN,XRMS
 10     FORMAT (" <6>"F9.0," VALUES IN "S4/
      1 " <6>"F9.0," VALUES <"F9.2/
      2 " <6>"F9.0," VALUES >"F9.2/
      4 " <6>MIN:"F9.2"  MAX:"F9.2"  MEAN:"F9.2"  RMS:"F9.2)
        RETURN
        END
R
```

```
TYPE PBSTG.FR
C       PROGRAM NAME: PBSTG1
        SUBROUTINE PBSTG          ;PLAYBACK,GATHER STATISTICS
        DOUBLE PRECISION CMEAN,RMS,UNIT,DTEMP
        COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1          ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
        COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
        COMMON /CSTAT/ NREC,CNUM(256),CMIN(256),CMAX(256),CMEAN(256),
     1          MIN(256),MAX(256),RMS(256)
        COMMON /CTREC/ JDATE(6),IMS,ITYP,IN(4088),NIN
        CALL ASUSP (2)
        NREC=NREC+1
        DO 110 I1=1,NIN,2
        IADR=IAND(IN(I1),777K)
        IVAL=IAND(IN(I1+1),1777K)
        CNUM(IADR)=CNUM(IADR)+1.
        IF (IVAL.LT.ITMIN(IADR)) CMIN(IADR)=CMIN(IADR)+1.
        IF (IVAL.GT.ITMAX(IADR)) CMAX(IADR)=CMAX(IADR)+1.
        IF (IVAL.LT.MIN(IADR)) MIN(IADR)=IVAL
        IF (IVAL.GT.MAX(IADR)) MAX(IADR)=IVAL
        VAL=PBVAL(IVAL,IADR,UNIT)
        IF(IADR.EQ.228.OR.IADR.EQ.229.OR.IADR.EQ.237) GO TO 110
        DTEMP=DBLE(VAL)
        CMEAN(IADR)=CMEAN(IADR)+DTEMP
        DTEMP=DTEMP*DTEMP
        RMS(IADR)=RMS(IADR)+DTEMP
        GO TO 110
C
C       COMPUTE RUDDER TORQUE    ;USE SPARE CHANNEL #254
        IF(IADR.EQ.200)RUDANG=VAL
        IF(IADR.NE.202)GO TO 109
        TEMP=VAL*(77964.-6.04*RUDANG-10.36*RUDANG*RUDANG)
        CMEAN(254)=CMEAN(254)+TEMP
        IF(CNUM(IADR).NE.1) GO TO 111
        CMIN(254)=TEMP
        CMAX(254)=TEMP
111     IF(TEMP.GT.CMAX(254)) CMAX(254)=TEMP
        IF(TEMP.LT.CMIN(254)) CMIN(254)=TEMP
        GO TO 110
C
C       COMPUTE PROP. SPINDLE TORQUE     ;USE SPARE CHANNEL #251
109     IF(IADR.EQ.201)PPICH=VAL
        IF(IADR.NE.203)GO TO 110
        TEMP=102150.*VAL*(1.+PPICH*PPICH*0.0003)
        CMEAN(251)=CMEAN(251)+TEMP
        IF(CNUM(IADR).NE.1) GO TO 112
        CMIN(251)=TEMP
        CMAX(251)=TEMP
112     IF(TEMP.GT.CMAX(251)) CMAX(251)=TEMP
        IF(TEMP.LT.CMIN(251)) CMIN(251)=TEMP
110     CONTINUE
        CALL ARDY(2)
        RETURN
        END
R
```

```
TYPE PBSTI.FR
      SUBROUTINE PBSTI          ;PLAYBACK, INITIALIZE STATISTICS
      DOUBLE PRECISION CMEAN,RMS
      COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1        ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
      COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
      COMMON /CSTAT/ NREC,CNUM(256),CMIN(256),CMAX(256),CMEAN(256),
     1        MIN(256),MAX(256),RMS(256)
      NREC=0
      DO 110 I1=1,IHIAD
      CNUM(I1)=0.
      CMIN(I1)=0.
      CMAX(I1)=0.
      CMEAN(I1)=0.
      MIN(I1)=1023
      MAX(I1)=0
      RMS(I1)=0.0
110   CONTINUE
      RETURN
      END
R
```

```
TYPE PBVAL.FR
C       PROGRAM NAME: PBVAL1
C
        FUNCTION PBVAL(IVAL,IADR,AUNIT)
C
C       CONVERSION TO ENGG. UNITS
C
        DOUBLE PRECISION UNIT,AUNIT
        COMMON /CVALUE/ ITYP(255),UNIT(15),ISUB(32)
C
        DATA ITYP /2*3,2*1,6*2,4*1,2*31,3*29,31,2*29,8*31,3/
        DATA ITYP(32) /29,27,2*26,31,27,29,2*31,2*32,26,4*27,26/
        DATA ITYP(49) /14*27,3*31,5*27,29,2*31,2*27,2*28,2*31,5*27/
        DATA ITYP(85) /32,4*27,32,15*27,4*32,8*27,2*32,6*27,31,2*27/
        DATA ITYP(129) /2*26,2*27,3*32,27,32,27,8*32,2*27,6*32,2*27/
        DATA ITYP(157) /5*27,30,3*27,3*32,27,4*32,26,12*1,2*4,9*25/
        DATA ITYP(198) /8,9,10,11,5,6,4*25,7,3*25/
        DATA ITYP(212) /3*3,9*25,13,2*14,25,3*15/
        DATA ITYP(231) /16,17,18,19,25,20,2*23,2*25/
        DATA ITYP(241) /21,2*25,11,33,6*25,2*22,2*25/
C
        DATA ISUB /1,2,3,4,5,5,6,7,8,9,9,12,10,11,4,2,12,13/
        DATA ISUB(19) /12,13,12,14,12,9,15,1,1,1,1,1,1,1,9/
C
        DATA UNIT(1) /"uS"/
        DATA UNIT(2) /"deg C"/
        DATA UNIT(3) /"g"/
        DATA UNIT(4) /"kPa"/
        DATA UNIT(5) /"MPa"/
        DATA UNIT(6) /"KNm"/
        DATA UNIT(7) /"tonnes"/
        DATA UNIT(8) /"%"/
        DATA UNIT(9) /"deg"/
        DATA UNIT(10) /"tenths"/
        DATA UNIT(11) /"tenths"/
        DATA UNIT (12) /" "/
        DATA UNIT(13) /"knots"/
        DATA UNIT(14) /"m"/
        DATA UNIT(15) /"V"/
C
        IF((IADR.LT.1).OR.(IADR.GT.255)) GO TO 500
        JTYP=ITYP(IADR)
        KTYP=ISUB(JTYP)
        AUNIT=UNIT(KTYP)
        IF(JTYP.GT.33) GO TO 250
        GO TO(10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,
     1  180,190,200,210,220,230,240,250,260,270,280,290,300,310,320,330),JTYP
C
C       1. STRAIN GAUGES GF=2.0 [ OTHER S/G's ]
10      PBVAL=IVAL*3.90625-2000.
        RETURN
C
C       2. THERMISTORS [5-10,171-172]
20      PBVAL=35.98-IVAL*0.1118
        RETURN

C       3. ACCELEROMETERS [1-2,31,212-214]
30      PBVAL=IVAL*0.0049-2.5
        RETURN
C
```

```
C     4. BUBBLER PRESSURE  [187-188]
40    PBVAL=IVAL*0.3359
      RETURN
C
C     5. RUDDER DIFFERENTIAL PRESSURE [202]
50    PBVAL=IVAL*0.0336
      RETURN
C
C     6. PROPELLER DIFFERENTIAL PRESSURE [203]
60    PBVAL=IVAL*1.348E-2
      RETURN
C
C     7. SHAFT TORQUE [208]
70    PBVAL=IVAL*0.515
      RETURN
C
C     8. THRUST [198]
80    PBVAL=IVAL*0.2344-120.
      RETURN
C
C     9. FUEL RACK [199]
90    PBVAL=IVAL*0.2201-9.97
      RETURN
C
C     10. RUDDER ANGLE RESPONSE [+ STBD, - PORT]  [200]
100   PBVAL=IVAL*0.07129-36.
      RETURN
C
C     11. PROPELLER RESPONSE [+ AHEAD,- ASTERN]  [201,244]
[illegible]0   PBVAL=IVAL*0.04688-22.
      RETURN
C
C     12. BUBBLER VALVE STATUS
120   RETURN
C
C     13. VISIBILITY [224]
130   ITEMP=IVAL*0.00977+0.5
      PBVAL=ITEMP
      RETURN
C
C     14. CLOUD AND ICE COVER  [225,226]
140   ITEMP=IVAL*0.00977+0.5
      PBVAL=ITEMP
      RETURN
C
C     15. BAROMETRIC PRESSURE  [228-230]
150   IF(IADR.EQ.229)GO TO 451
      IF(IADR.EQ.230)GO TO 452
      IBARO1=IVAL*0.00977+0.5
      RETURN
451   IBARO2=IVAL*0.00977+0.5
      RETURN
452   IBARO=IBARO1+10*IBARO2+900
      IF(IVAL.GT.922)IBARO=IBARO+100
      PBVAL=IBARO/10.
      RETURN
C
C     16. AIR TEMPERATURE [231]
160   PBVAL=IVAL*0.19531-100.
      RETURN
```

```
C
C       17. RELATIVE HUMIDITY [232]
170     PBVAL=IVAL*0.09766
        RETURN
C
C       18. WIND SPEED [233]
180     PBVAL=0.1357*IVAL
        RETURN
C
C       19. WIND DIRECTION [234]
190     IF(IVAL.GT.841) IVAL=1024-IVAL
        PBVAL=296.-IVAL*0.3516
        RETURN
C
C       20. SHIP SPEED [236]
200     PBVAL=0.03*IVAL
        RETURN
C
C       21. ENGINE RPM [241]
210     PBVAL=IVAL*0.5547-4.
        RETURN
C
C       22. DRAUGHT [252-253]
220     ITEMP=0.00977*IVAL+0.5
        PBVAL=0.5*ITEMP+7.
        RETURN
C
C       23. SHIP'S HEADING [237-238]
230     TEMP2=(IVAL-512)*0.0195
        IF (IADR.EQ.238) GO TO 231
        TEMP1=TEMP2
        PBVAL=0.0
        RETURN
231     PBVAL=57.296*ATAN2(TEMP1,TEMP2)
240     RETURN
C
C       25. OTHER CHANNELS
C       [189-197,204-211,215-223,227,233,235,239-240,242-243,246-251,254-255]
250     PBVAL=IVAL*0.01963-10.
        RETURN
C
C       26. STRAIN GAUGES GF=1.77 [34-35,43,48,129,155,174]
260     PBVAL=(IVAL-512)*3.4570
        RETURN
C
C       27. STRAIN GAUGES GF=1.83 [33, etc. ]
270     PBVAL=(IVAL-512)*3.5742
        RETURN
C
C       28. STRAIN GAUGES GF=1.86 [76]
280     PBVAL=(IVAL-512)*3.6328
        RETURN
C
C       29. STRAIN GAUGES GF=1.90 [17, etc. ]
290     PBVAL=(IVAL-512)*3.7109
        RETURN
C
C       30. STRAIN GAUGES GF=1.94 [162]
300     PBVAL=(IVAL-512)*3.7891
        RETURN
```

```
C
C       31. STRAIN GAUGES GF=1.95 [ 15, etc. ]
310     PBVAL=(IVAL-512)*3.8086
        RETURN
C
C       32. STRAIN GAUGES GF=1.96 [ 41-42, etc. ]
320     PBVAL=(IVAL-512)*3.8281
        RETURN
C
C       33. RUDDER ANGLE COMMAND [245]
330     PBVAL=IVAL*0.0795-40.7
        RETURN
C
C
500     WRITE (ICTTO,501) IADR,IVAL
501     FORMAT(" <6>BAD SENSOR: IADR= ",I3,"  IVAL=",I3)
        RETURN
C
        END
```

```
TYPE PBSTT.FR
C       PROGRAM NAME: PBSTT1
        SUBROUTINE PBSTT          ;PLAYBACK, TYPE STATISTICS
        DOUBLE PRECISION UNIT,CMEAN,RMS
        COMMON /CHEAD/ IDATE(6),ILG,IRM,IHIAD,IDUM,
     1          ITAB1(0:511),ITMIN(0:511),ITMAX(0:511)
        COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
        COMMON /CSTAT/ NREC,CNUM(256),CMIN(256),CMAX(256),CMEAN(256),
     1          MIN(256),MAX(256),RMS(256)
        COMMON /CTPST/ LADR1,LADR2,LINCR,LTFOR,LPFOR,LALL
   20   FORMAT (" <6>ADR R/M    MIN.AL   MAX.AL    #REC    %<MIN"Z)
   21   FORMAT (1X"    %>MAX       MIN       MAX      MEAN       RMS        SD")
   22   FORMAT (" <6>"I3,I2,1XA2,2F9.2,F9.0,2F9.2Z)
   23   FORMAT(1X,5F9.2,1X,S8)
   25   FORMAT (" <6>ADR R/M     #REC       MIN       MAX      MEAN ",
    1   "     RMS        SD")
   26   FORMAT (" <6>"I3,I2,1XA2,F9.0,5F9.2,1X,S8)
   11   FORMAT (A2)
   12   FORMAT (" <6>NO NO"/" <6>YE YES"/" <6>YL LONG FORMAT"/
    1     " <6>YS SHORT FORMAT"/
    2     " <6>AL ALL SENSORS, LONG FORMAT"/
    3     " <6>AS ALL SENSORS, SHORT FORMAT")
   17   FORMAT (" <6>"I6" TAPE RECORDS READ.")
C  INITIALIZE
        WRITE (ICTTO,17) NREC
        LPFOR=0
        LTFOR=0
        LADR1=1
        LADR2=IHIAD
        LINCR=1
C  ASK
110     LALL=1
        WRITE(ICTTO,10)
   10   FORMAT(" <6>TYPE STATISTICS? "Z)
        READ (ICTTI,11) IX
        IF (IX.EQ."NO")RETURN
        IF (IX.EQ."AS") GO TO 170
        IF (IX.EQ."AL") GO TO 180
        LALL=0
        IF (IX.EQ."YE") GO TO 140
        IF (IX.EQ."YS") GO TO 120
        IF (IX .EQ. "YL") GO TO 130
C
        WRITE (ICTTO,12)
        GO TO 110
C
120     LPFOR=-1
        GO TO 140
130     LPFOR=1
C
C   GET RANGE
140     WRITE (ICTTO,14)
   14   FORMAT (" <6>ENTER 'ADR1,ADR2', OR '0,0' TO EXIT: "Z)
144     READ (ICTTI,ERR=146,END=400) LADR1,LADR2
        IF (LADR1.EQ.LADR2.AND.LADR1.EQ.0) GO TO 400
        IF (LADR1.LE.0.OR.LADR2.LT.LADR1) GO TO 146
        GO TO 150
146     WRITE (ICTTO,15)
   15   FORMAT(" <6>?: "Z)
```

```
        GO TO 144
C   DETERMINE FORMAT
150     LTFOR=LPFOR
        IF (LTFOR) 170,160,180
  0     WRITE(ICTTO,16)
  16    FORMAT(" <6>LONG OR SHORT? "Z)
        READ (ICTTI,18,END=400) IX
  18    FORMAT (A1)
        IF (IX.EQ."S ") GO TO 170
        IF (IX.EQ."L ")GO TO 180
        GO TO 160
C   SHORT FORMAT HEADER
170     WRITE (ICTTO,25)
        LTFOR = -1
        GO TO 200
C   LONG FORMAT HEADER
180     WRITE (ICTTO,20)
        WRITE (ICTTO,21)
        LTFOR=1
C
C   LOOP TO TYPE DATA
C
200     DO 250 I1=LADR1,LADR2,LINCR
        IF (CNUM(I1).EQ.0) GO TO 250
        IF(I1.EQ.228.OR.I1.EQ.229.OR.I1.EQ.237)GO TO 250
        IRAT=ISHFT(IAND(16000K,ITAB1(I1)),-10)
        IF(IRAT.EQ.0) GO TO 250
        IM="P "
        IF (IAND(1000K,ITAB1(I1))) IM="R "
        IF(I1.EQ.237)GO TO 250      ;SKIP FOR HDG(SIN)
        XMEAN=SNGL(CMEAN(I1)/DBLE(CNUM(I1)))
        XRMS=SNGL(RMS(I1)/DBLE(CNUM(I1)))
        XRMS=SQRT(XRMS)
        IF(MIN(I1).EQ.MAX(I1))GO TO 201
        IF((I1.GE.224).AND.(I1.LE.230))GO TO 202        ;AVOID ERROR 16
        IF(I1.GE.252)GO TO 202                          ;FOR QUANTIZED VALUES
        XSD=SNGL((RMS(I1)-CMEAN(I1)*DBLE(XMEAN))/DBLE(CNUM(I1)-1.))
        XSD=SQRT(XSD)
        GO TO 202
201     XSD=0.0 ;AVOID NUMERICAL ERRORS
202     PMIN=100.*CMIN(I1)/CNUM(I1)
        PMAX=100.*CMAX(I1)/CNUM(I1)
        PTHR=0.01
        IF(CMIN(I1).NE.0.AND.PMIN.LE.PTHR)PMIN=0.01
        IF(CMAX(I1).NE.0.AND.PMAX.LE.PTHR)PMAX=0.01
        XMIN=PBVAL(MIN(I1),I1,UNIT)
        XMAX=PBVAL(MAX(I1),I1,UNIT)
        IF (LTFOR) 210,250,220
C   SHORT FORMAT
210     WRITE (ICTTO,26) I1,IRAT,IM,CNUM(I1),XMIN,XMAX,XMEAN,XRMS,XSD,UNIT
        GO TO 250
C   LONG FORMAT
220     AMIN=PBVAL(ITMIN(I1),I1,UNIT)
        AMAX=PBVAL(ITMAX(I1),I1,UNIT)
        WRITE (ICTTO,22) I1,IRAT,IM,AMIN,AMAX,CNUM(I1),PMIN,PMAX
        WRITE (ICTTO,23) XMIN,XMAX,XMEAN,XRMS,XSD,UNIT
 50     CONTINUE

        IF(LALL.EQ.1)RETURN
        LTFOR=0
        GO TO 140
400     TYPE "<6>"
        RETURN
        END
```

```
TYPE IPBYN.FR
        FUNCTION IPBYN(I)
        COMMON /CHANL/ ICTTO,ICTTI,ICMT0,ICMT1,JREAD,ISTS,IER,NWORD
  0     READ (ICTTI,11) IYN
 11     FORMAT (S1)
        IF (IYN .EQ. "Y") GO TO 120
        IF (IYN .EQ. "N") GO TO 130
        WRITE(ICTTO,12)
 12     FORMAT(" <6>ANSWER Y OR N "Z)
        GO TO 110
120     IPBYN=1
        RETURN
130     IPBYN=0
        RETURN
        END
R
```

# APPENDIX C

## SOFTWARE BACK-UP

## APPENDIX C

### SOFTWARE BACK-UP

Good programming practice requires back-up copies to be made of all original software. Stated simply:

ORIGINAL VERSIONS MUST NEVER BE
USED FOR SOFTWARE DEVELOPMENT.

While working on software development, periodic updates to the current version should be made to ensure that in the event of a disk and/or hardware failure, only a small amount of labour is lost.

All original ROLM 1603A Software, as well as programs developed specially for this system have been stored on a tape labelled 'BACK-UP TAPE'. The tape files contain the following programs:

| Tape File No. | Developed by | Contents | | | |
|---|---|---|---|---|---|
| 0: | IVL | "L" Program<br>Original Data Acquisition<br>and Playback Programs<br>April 1979 | | | |
| 1 | IVL | "L" Program Listings<br>Source versions of Data Acquisition<br>and Playback Programs<br>April 1979 | | | |
| 2 | ROLM | P300 | RDOS | 6.50 | 1 of 3 |
| 3 | ROLM | | RDOS | 6.50 | 2 of 3 |
| 4 | ROLM | | RDOS | 6.50 | 3 of 3 |
| 5 | ROLM | P203 | RTOS | 5.50 | |
| 6 | G & M | Data Acquisition Program (source and binary versions, other related programs)<br>November 1980 | | | |
| 7 | G & M | Playback Program (source and binary versions, other related programs)<br>November 1980 | | | |

Disks are saved as files on magnetic tape using the DUMP command. Dumped files may be retrieved onto disk with the LOAD command.

# APPENDIX D

## SYSTEM DIAGNOSTICS

APPENDIX D

## SYSTEM DIAGNOSTICS

Diagnostic Software to test proper functioning of the System Hardware has been made available by ROLM Corporation. The tests for each module are performed independently, thereby easing detection of a faulty one, if any. Diagnostics are stored on diskette and the corresponding literature is available. The following tests are necessary for the current system configuration.

| Program | Function |
|---|---|
| TERMD | Teletype Test |
| FLDD | Floppy Disk Test |
| EXER | EAU Diagnostic Exercises |
| MAT | Memory Address Tester |
| CBD | Checkerboard Data Test |
| PWR03 | Power Monitor Autorestart Test |
| INS03 | CPU Logic Test |
| MTA1R | Magnetic Tape Exercises |
| SRT03 | System Reliability Test |

# APPENDIX E

## ROLM DOCUMENTATION

APPENDIX E

## ROLM DOCUMENTATION

The following documentation has been made available by ROLM Corporation:

| Publication No. | Title |
|---|---|
| - | User's Manual for the ROLM Models 1603 and 1603A (AN/UYK-27) Processor |
| 493-100188-01 | How to Load and Generate Your RDOS on RMX/RDOS System |
| 493-100083-05 | Introduction to RDOS |
| 493-100075-08 | Real-Time Disk Operating System - Reference Manual |
| 493-100093-02 | Introduction to Real Time Operating System |
| 493-100056-07 | The 1603 Real-Time Operating System - Reference Manual |
| 493-100109-03 | RDOS - RMX/RDOS - Command Line Interpreter Reference Manual |
| 493-100018-08 | Text Editor - User's Manual |
| 493-100111-00 | Superedit User's Manual |
| 493-100080-05 | Relocatable Trader User's Manual |
| 493-100081-05 | User's Manual Macro Assembler |
| 493-100040-01 | Extended Assembler - User's Manual |
| 493-100074-04 | User's Manual - Library File Editor |
| 493-100053-08 | User's Manual Fortran IV |
| 493-100088-05 | Run Time Library User's Manual Fortran IV |
| 493-105021-02 | RDOS External Memory Software Support |

| Publication No. | Title |
|---|---|
| 493-100044-04 | 1603 Debugger User's Manual |
| 493-100186-01 | Diskedit User's Manual |
| 493-100084-02 | Octal Editor User's Manual |
| 493-100052-05 | Extended ALGOL User's Manual |
| 493-100065-06 | Extended Basic User's Manual |
| 493-100119-01 | Extended Basic System Manager's Guide |
| 493-100105-03 | RDOS - RMX/RDOS User's Handbook |
| 497-100009-04 | Diagnostic Program - Teletype Test II |
| 497-100004-04 | EAU Diagnostic Program Exercised |
| 497-02040-01 | Diagnostic Program<br>1602 Memory Address Tester |
| 497-002041-05 | Diagnostic Program<br>Checkerboard Data Test |
| 497-105018-00 | 1603 Power Monitor, Autorestart Test |
| 497-105034-00 | 1603 CPU Logic Test |
| 497-150311 | System Reliability Test SRT03 |